

# API Documentation for Jeliot 3 and DynamicJava Classes

March 7, 2004

# Contents

<b>1</b>	<b>Package jeliot.lang</b>	<b>12</b>
1.1	Classes . . . . .	13
1.1.1	CLASS <b>ArrayInstance</b> . . . . .	13
1.1.2	CLASS <b>ArrayUtilities</b> . . . . .	16
1.1.3	CLASS <b>ClassInfo</b> . . . . .	17
1.1.4	CLASS <b>Instance</b> . . . . .	19
1.1.5	CLASS <b>MethodFrame</b> . . . . .	21
1.1.6	CLASS <b>ObjectFrame</b> . . . . .	23
1.1.7	CLASS <b>Reference</b> . . . . .	25
1.1.8	CLASS <b>Value</b> . . . . .	27
1.1.9	CLASS <b>Variable</b> . . . . .	29
1.1.10	CLASS <b>VariableInArray</b> . . . . .	30
<b>2</b>	<b>Package koala.dynamicjava.parser</b>	<b>32</b>
2.1	Interfaces . . . . .	34
2.1.1	INTERFACE <b>Parser.CompileTimeExpression</b> . . . . .	34
2.1.2	INTERFACE <b>ParserConstants</b> . . . . .	34
2.2	Classes . . . . .	41
2.2.1	CLASS <b>ASCII_UCodeESC_CharStream</b> . . . . .	41
2.2.2	CLASS <b>ParseException</b> . . . . .	44
2.2.3	CLASS <b>Parser</b> . . . . .	47
2.2.4	CLASS <b>Parser.ArgumentsSuffix</b> . . . . .	106
2.2.5	CLASS <b>Parser.ArrayReferenceSuffix</b> . . . . .	107
2.2.6	CLASS <b>Parser.ClassBody</b> . . . . .	108
2.2.7	CLASS <b>Parser.CompileTimeQualified Name</b> . . . . .	108
2.2.8	CLASS <b>Parser.CompileTimeSuperExpression</b> . . . . .	113
2.2.9	CLASS <b>Parser.DotAllocationSuffix</b> . . . . .	117
2.2.10	CLASS <b>Parser.DotIdentifierSuffix</b> . . . . .	118
2.2.11	CLASS <b>Parser.DotThisSuffix</b> . . . . .	119
2.2.12	CLASS <b>Parser.ExpressionSuffix</b> . . . . .	120
2.2.13	CLASS <b>Parser.JJCalls</b> . . . . .	121
2.2.14	CLASS <b>Parser.ModifierFlags</b> . . . . .	121
2.2.15	CLASS <b>Parser.TokenManager</b> . . . . .	121
2.2.16	CLASS <b>Token</b> . . . . .	124
2.2.17	CLASS <b>TokenMgrError</b> . . . . .	126
<b>3</b>	<b>Package koala.dynamicjava.parser.wrapper</b>	<b>129</b>
3.1	Interfaces . . . . .	130
3.1.1	INTERFACE <b>ParserFactory</b> . . . . .	130

3.1.2	INTERFACE <b>SourceCodeParser</b> . . . . .	130
3.2	Classes . . . . .	131
3.2.1	CLASS <b>JavaCCParser</b> . . . . .	131
3.2.2	CLASS <b>JavaCCParserFactory</b> . . . . .	133
3.2.3	CLASS <b>ParseError</b> . . . . .	134
3.2.4	CLASS <b>TreeToken</b> . . . . .	136
<b>4</b>	<b>Package jeliot.launcher</b> . . . . .	<b>139</b>
4.1	Classes . . . . .	140
4.1.1	CLASS <b>Launcher</b> . . . . .	140
<b>5</b>	<b>Package koala.dynamicjava.tree.visitor</b> . . . . .	<b>144</b>
5.1	Interfaces . . . . .	145
5.1.1	INTERFACE <b>Visitor</b> . . . . .	145
5.2	Classes . . . . .	160
5.2.1	CLASS <b>VisitorObject</b> . . . . .	160
<b>6</b>	<b>Package jeliot.gui</b> . . . . .	<b>176</b>
6.1	Classes . . . . .	177
6.1.1	CLASS <b>AboutWindow</b> . . . . .	177
6.1.2	CLASS <b>CodeEditor</b> . . . . .	199
6.1.3	CLASS <b>CodePane</b> . . . . .	228
6.1.4	CLASS <b>DraggableComponent</b> . . . . .	252
6.1.5	CLASS <b>HelpWindow</b> . . . . .	277
6.1.6	CLASS <b>JavaFileFilter</b> . . . . .	299
6.1.7	CLASS <b>JeliotWindow</b> . . . . .	300
6.1.8	CLASS <b>LineNumbers</b> . . . . .	308
6.1.9	CLASS <b>LoadJeliot</b> . . . . .	332
6.1.10	CLASS <b>OutputConsole</b> . . . . .	333
6.1.11	CLASS <b>TheaterPopup</b> . . . . .	362
<b>7</b>	<b>Package koala.dynamicjava.interpreter</b> . . . . .	<b>365</b>
7.1	Interfaces . . . . .	368
7.1.1	INTERFACE <b>ClassLoaderContainer</b> . . . . .	368
7.1.2	INTERFACE <b>Interpreter</b> . . . . .	368
7.2	Classes . . . . .	372
7.2.1	CLASS <b>ClassFactory</b> . . . . .	372
7.2.2	CLASS <b>ClassInfoCompiler</b> . . . . .	383
7.2.3	CLASS <b>ClassInfoCompiler.ConstructorVisitor</b> . . . . .	384
7.2.4	CLASS <b>ClassInfoCompiler.MembersVisitor</b> . . . . .	408
7.2.5	CLASS <b>ClassPool</b> . . . . .	422
7.2.6	CLASS <b>EvaluationVisitor</b> . . . . .	423
7.2.7	CLASS <b>InterpreterException</b> . . . . .	451
7.2.8	CLASS <b>InterpreterException.SourceInformation</b> . . . . .	453
7.2.9	CLASS <b>InterpreterUtilities</b> . . . . .	454
7.2.10	CLASS <b>InterpreterUtilities.AddOperation</b> . . . . .	461
7.2.11	CLASS <b>InterpreterUtilities.BinaryArithmeticOperation</b> . . . . .	462
7.2.12	CLASS <b>InterpreterUtilities.BinaryPredicate</b> . . . . .	463
7.2.13	CLASS <b>InterpreterUtilities.BitAndOperation</b> . . . . .	463
7.2.14	CLASS <b>InterpreterUtilities.BitOrOperation</b> . . . . .	464

7.2.15	CLASS	<b>InterpreterUtilities.BitwiseOperation</b>	465
7.2.16	CLASS	<b>InterpreterUtilities.DivideOperation</b>	465
7.2.17	CLASS	<b>InterpreterUtilities.EqualToPredicate</b>	466
7.2.18	CLASS	<b>InterpreterUtilities.GreaterOrEqualOperation</b>	467
7.2.19	CLASS	<b>InterpreterUtilities.GreaterThanOperation</b>	468
7.2.20	CLASS	<b>InterpreterUtilities.LessOrEqualOperation</b>	468
7.2.21	CLASS	<b>InterpreterUtilities.LessThanOperation</b>	469
7.2.22	CLASS	<b>InterpreterUtilities.MinusOperation</b>	469
7.2.23	CLASS	<b>InterpreterUtilities.MultiplyOperation</b>	470
7.2.24	CLASS	<b>InterpreterUtilities.NotEqualToPredicate</b>	471
7.2.25	CLASS	<b>InterpreterUtilities.PlusOperation</b>	472
7.2.26	CLASS	<b>InterpreterUtilities.RelationalOperation</b>	473
7.2.27	CLASS	<b>InterpreterUtilities.RemainderOperation</b>	473
7.2.28	CLASS	<b>InterpreterUtilities.ShiftLeftOperation</b>	474
7.2.29	CLASS	<b>InterpreterUtilities.ShiftOperation</b>	475
7.2.30	CLASS	<b>InterpreterUtilities.ShiftRightOperation</b>	475
7.2.31	CLASS	<b>InterpreterUtilities.SubtractOperation</b>	476
7.2.32	CLASS	<b>InterpreterUtilities.UnaryOperation</b>	477
7.2.33	CLASS	<b>InterpreterUtilities.UnsignedShiftRightOperation</b>	478
7.2.34	CLASS	<b>InterpreterUtilities.XOrOperation</b>	479
7.2.35	CLASS	<b>Main</b>	479
7.2.36	CLASS	<b>NameVisitor</b>	480
7.2.37	CLASS	<b>NodeProperties</b>	506
7.2.38	CLASS	<b>TreeClassFinder</b>	508
7.2.39	CLASS	<b>TreeClassLoader</b>	510
7.2.40	CLASS	<b>TreeCompiler</b>	515
7.2.41	CLASS	<b>TreeCompiler.ClassInfoLoader</b>	516
7.2.42	CLASS	<b>TreeCompiler.CompilationUnitVisitor</b>	519
7.2.43	CLASS	<b>TreeCompiler.PseudoError</b>	533
7.2.44	CLASS	<b>TreeInterpreter</b>	535
7.2.45	CLASS	<b>TreeInterpreter.ConstructorParametersDescriptor</b>	542
7.2.46	CLASS	<b>TreeInterpreter.MethodDescriptor</b>	543
7.2.47	CLASS	<b>TypeChecker</b>	543
7.2.48	CLASS	<b>UninitializedObject</b>	571
<b>8</b>		<b>Package koala.dynamicjava.util</b>	<b>572</b>
8.1	Classes		573
8.1.1	CLASS	<b>AmbiguousFieldException</b>	573
8.1.2	CLASS	<b>BufferedImportationManager</b>	574
8.1.3	CLASS	<b>Constants</b>	577
8.1.4	CLASS	<b>DisplayVisitor</b>	578
8.1.5	CLASS	<b>FileFinder</b>	606
8.1.6	CLASS	<b>ImportationManager</b>	607
8.1.7	CLASS	<b>LibraryFinder</b>	610
8.1.8	CLASS	<b>LocalizedMessageReader</b>	611
8.1.9	CLASS	<b>ReflectionUtilities</b>	612

<b>9</b>	<b>Package jeliot.mcode</b>	<b>616</b>
9.1	Classes	617
9.1.1	CLASS <b>Code</b>	617
9.1.2	CLASS <b>Command</b>	622
9.1.3	CLASS <b>Interpreter</b>	623
9.1.4	CLASS <b>InterpreterError</b>	627
9.1.5	CLASS <b>MCodeUtilities</b>	628
<b>10</b>	<b>Package koala.dynamicjava.interpreter.throwable</b>	<b>636</b>
10.1	Classes	637
10.1.1	CLASS <b>BreakException</b>	637
10.1.2	CLASS <b>ContinueException</b>	639
10.1.3	CLASS <b>ReturnException</b>	641
10.1.4	CLASS <b>ThrownException</b>	644
<b>11</b>	<b>Package koala.dynamicjava.interpreter.context</b>	<b>647</b>
11.1	Interfaces	649
11.1.1	INTERFACE <b>Context</b>	649
11.1.2	INTERFACE <b>SimpleContext</b>	655
11.2	Classes	658
11.2.1	CLASS <b>GlobalContext</b>	658
11.2.2	CLASS <b>GlobalContext.CompilationUnitVisitor</b>	668
11.2.3	CLASS <b>GlobalContext.CompilationUnitVisitor.MembersVisitor</b>	683
11.2.4	CLASS <b>GlobalContext.PseudoClassLoader</b>	696
11.2.5	CLASS <b>GlobalContext.PseudoError</b>	699
11.2.6	CLASS <b>MethodContext</b>	701
11.2.7	CLASS <b>MethodModificationError</b>	714
11.2.8	CLASS <b>NoSuchFunctionException</b>	716
11.2.9	CLASS <b>StaticContext</b>	718
11.2.10	CLASS <b>VariableContext</b>	730
11.2.11	CLASS <b>VariableContext.AbstractVariable</b>	733
11.2.12	CLASS <b>VariableContext.Constant</b>	734
11.2.13	CLASS <b>VariableContext.Link</b>	735
11.2.14	CLASS <b>VariableContext.LinkFactory</b>	736
11.2.15	CLASS <b>VariableContext.Scope</b>	736
11.2.16	CLASS <b>VariableContext.Scope.Entry</b>	738
11.2.17	CLASS <b>VariableContext.Scope.EntryFactory</b>	738
11.2.18	CLASS <b>VariableContext.Variable</b>	739
<b>12</b>	<b>Package koala.dynamicjava.classinfo</b>	<b>741</b>
12.1	Interfaces	743
12.1.1	INTERFACE <b>ClassFinder</b>	743
12.1.2	INTERFACE <b>ClassInfo</b>	744
12.1.3	INTERFACE <b>ConstructorInfo</b>	746
12.1.4	INTERFACE <b>FieldInfo</b>	747
12.1.5	INTERFACE <b>MethodInfo</b>	747
12.2	Classes	748
12.2.1	CLASS <b>ClassInfoUtilities</b>	748
12.2.2	CLASS <b>JavaClassInfo</b>	751
12.2.3	CLASS <b>JavaConstructorInfo</b>	755

12.2.4	CLASS	<b>JavaFieldInfo</b>	756
12.2.5	CLASS	<b>JavaMethodInfo</b>	757
12.2.6	CLASS	<b>TreeClassInfo</b>	759
12.2.7	CLASS	<b>TreeClassInfo.MembersVisitor</b>	764
12.2.8	CLASS	<b>TreeConstructorInfo</b>	778
12.2.9	CLASS	<b>TreeFieldInfo</b>	780
12.2.10	CLASS	<b>TreeMethodInfo</b>	781
12.2.11	CLASS	<b>TypeVisitor</b>	784
<b>13</b>	<b>Package</b>	<b>koala.dynamicjava.gui</b>	<b>799</b>
13.1	Interfaces		802
13.1.1	INTERFACE	<b>MessageHandler</b>	802
13.2	Classes		803
13.2.1	CLASS	<b>Editor</b>	803
13.2.2	CLASS	<b>Editor.DocumentAdapter</b>	831
13.2.3	CLASS	<b>Editor.OpenAction</b>	832
13.2.4	CLASS	<b>Editor.RedoAction</b>	833
13.2.5	CLASS	<b>Editor.SaveAction</b>	834
13.2.6	CLASS	<b>Editor.SaveAsAction</b>	835
13.2.7	CLASS	<b>Editor.UndoAction</b>	836
13.2.8	CLASS	<b>Editor.UndoHandler</b>	837
13.2.9	CLASS	<b>JTextComponentOutputStream</b>	838
13.2.10	CLASS	<b>Main</b>	839
13.2.11	CLASS	<b>Main.AboutAction</b>	864
13.2.12	CLASS	<b>Main.ClearAction</b>	865
13.2.13	CLASS	<b>Main.EditorCaretListener</b>	866
13.2.14	CLASS	<b>Main.EvalAction</b>	866
13.2.15	CLASS	<b>Main.EvalSelectionAction</b>	868
13.2.16	CLASS	<b>Main.ExitAction</b>	869
13.2.17	CLASS	<b>Main.InterpreterThread</b>	870
13.2.18	CLASS	<b>Main.OptionsAction</b>	872
13.2.19	CLASS	<b>Main.ReinitAction</b>	873
13.2.20	CLASS	<b>Main.ScrollBarModelChangeListener</b>	874
13.2.21	CLASS	<b>Main.StopAction</b>	875
13.2.22	CLASS	<b>OptionsDialog</b>	876
13.2.23	CLASS	<b>OptionsDialog.CancelButtonAction</b>	900
13.2.24	CLASS	<b>OptionsDialog.CPLAddButtonAction</b>	901
13.2.25	CLASS	<b>OptionsDialog.GUIPanel</b>	902
13.2.26	CLASS	<b>OptionsDialog.GUIPanel.CheckBoxChangeListener</b>	926
13.2.27	CLASS	<b>OptionsDialog.GUIPanel.FileCheckBoxChangeListener</b>	927
13.2.28	CLASS	<b>OptionsDialog.GUIPanel.InitFileBrowseButtonAction</b>	927
13.2.29	CLASS	<b>OptionsDialog.InterpreterPanel</b>	928
13.2.30	CLASS	<b>OptionsDialog.InterpreterPanel.CheckBoxChangeListener</b>	953
13.2.31	CLASS	<b>OptionsDialog.InterpreterPanel.FileCheckBoxChangeListener</b>	953
13.2.32	CLASS	<b>OptionsDialog.InterpreterPanel.InitFileBrowseButtonAction</b>	954
13.2.33	CLASS	<b>OptionsDialog.LPLAddButtonAction</b>	955
13.2.34	CLASS	<b>OptionsDialog.OKButtonAction</b>	956
13.2.35	CLASS	<b>OptionsDialog.OptionSet</b>	957
13.2.36	CLASS	<b>OptionsDialog.UCOKButtonAction</b>	958
13.2.37	CLASS	<b>StatusBar</b>	959

13.2.38	CLASS	<b>StatusBar.DisplayThread</b>	983
13.2.39	CLASS	<b>StringList</b>	985
13.2.40	CLASS	<b>StringList.DownButtonAction</b>	1010
13.2.41	CLASS	<b>StringList.ListSelectionAdapter</b>	1011
13.2.42	CLASS	<b>StringList.RemoveButtonAction</b>	1011
13.2.43	CLASS	<b>StringList.UpButtonAction</b>	1012
13.2.44	CLASS	<b>URLChooser</b>	1013
13.2.45	CLASS	<b>URLChooser.BrowseButtonAction</b>	1035
13.2.46	CLASS	<b>URLChooser.CancelButtonAction</b>	1036
13.2.47	CLASS	<b>URLChooser.ClearButtonAction</b>	1037
13.2.48	CLASS	<b>URLChooser.DocumentAdapter</b>	1038
13.2.49	CLASS	<b>URLChooser.OKButtonAction</b>	1039
<b>14</b>	<b>Package</b>	<b>koala.dynamicjava.classfile</b>	<b>1041</b>
14.1	Classes		1043
14.1.1	CLASS	<b>AbstractMethodIdentifier</b>	1043
14.1.2	CLASS	<b>AttributeInfo</b>	1044
14.1.3	CLASS	<b>AttributeOwnerComponent</b>	1045
14.1.4	CLASS	<b>BytecodeComponent</b>	1046
14.1.5	CLASS	<b>ClassFile</b>	1047
14.1.6	CLASS	<b>ClassIdentifier</b>	1050
14.1.7	CLASS	<b>CodeAttribute</b>	1051
14.1.8	CLASS	<b>CodeAttribute.ExceptionTableEntry</b>	1053
14.1.9	CLASS	<b>ConstantPool</b>	1054
14.1.10	CLASS	<b>ConstantPool.ClassInfo</b>	1058
14.1.11	CLASS	<b>ConstantPool.DoubleInfo</b>	1059
14.1.12	CLASS	<b>ConstantPool.FieldInfo</b>	1060
14.1.13	CLASS	<b>ConstantPool.FloatInfo</b>	1061
14.1.14	CLASS	<b>ConstantPool.Info</b>	1062
14.1.15	CLASS	<b>ConstantPool.IntegerInfo</b>	1062
14.1.16	CLASS	<b>ConstantPool.InterfaceMethodInfo</b>	1063
14.1.17	CLASS	<b>ConstantPool.LongInfo</b>	1064
14.1.18	CLASS	<b>ConstantPool.MethodInfo</b>	1065
14.1.19	CLASS	<b>ConstantPool.NameAndTypeInfo</b>	1066
14.1.20	CLASS	<b>ConstantPool.NameAndTypeKey</b>	1067
14.1.21	CLASS	<b>ConstantPool.StringInfo</b>	1067
14.1.22	CLASS	<b>ConstantPool.UTF8Info</b>	1068
14.1.23	CLASS	<b>ConstantString</b>	1069
14.1.24	CLASS	<b>ConstantValueAttribute</b>	1070
14.1.25	CLASS	<b>ExceptionsAttribute</b>	1072
14.1.26	CLASS	<b>FieldIdentifier</b>	1074
14.1.27	CLASS	<b>FieldInfo</b>	1075
14.1.28	CLASS	<b>InnerClassesAttribute</b>	1077
14.1.29	CLASS	<b>InnerClassesEntry</b>	1079
14.1.30	CLASS	<b>InterfaceMethodIdentifier</b>	1080
14.1.31	CLASS	<b>JVMUtilities</b>	1082
14.1.32	CLASS	<b>MemberIdentifier</b>	1083
14.1.33	CLASS	<b>MethodIdentifier</b>	1085
14.1.34	CLASS	<b>MethodInfo</b>	1086
14.1.35	CLASS	<b>SimpleAttribute</b>	1089

<b>15 Package koala.dynamicjava.interpreter.modifier</b>	<b>1091</b>
15.1 Classes . . . . .	1092
15.1.1 CLASS <b>ArrayModifier</b> . . . . .	1092
15.1.2 CLASS <b>FinalVariableModifier</b> . . . . .	1093
15.1.3 CLASS <b>InvalidModifier</b> . . . . .	1094
15.1.4 CLASS <b>LeftHandSideModifier</b> . . . . .	1095
15.1.5 CLASS <b>ObjectFieldModifier</b> . . . . .	1096
15.1.6 CLASS <b>StaticFieldModifier</b> . . . . .	1097
15.1.7 CLASS <b>SuperFieldModifier</b> . . . . .	1098
15.1.8 CLASS <b>VariableModifier</b> . . . . .	1099
<b>16 Package jeliot.theater</b>	<b>1101</b>
16.1 Interfaces . . . . .	1104
16.1.1 INTERFACE <b>ActorContainer</b> . . . . .	1104
16.1.2 INTERFACE <b>Controlled</b> . . . . .	1104
16.2 Classes . . . . .	1105
16.2.1 CLASS <b>ACActor</b> . . . . .	1105
16.2.2 CLASS <b>Actor</b> . . . . .	1109
16.2.3 CLASS <b>ActorFactory</b> . . . . .	1115
16.2.4 CLASS <b>AnimatingActor</b> . . . . .	1122
16.2.5 CLASS <b>Animation</b> . . . . .	1127
16.2.6 CLASS <b>AnimationEngine</b> . . . . .	1130
16.2.7 CLASS <b>Animator</b> . . . . .	1132
16.2.8 CLASS <b>ArrayActor</b> . . . . .	1134
16.2.9 CLASS <b>BubbleActor</b> . . . . .	1139
16.2.10 CLASS <b>CastActor</b> . . . . .	1144
16.2.11 CLASS <b>ComponentDragger</b> . . . . .	1149
16.2.12 CLASS <b>ConstantBox</b> . . . . .	1150
16.2.13 CLASS <b>Director</b> . . . . .	1154
16.2.14 CLASS <b>Director.InputAnimator</b> . . . . .	1168
16.2.15 CLASS <b>ExpressionActor</b> . . . . .	1169
16.2.16 CLASS <b>Highlight</b> . . . . .	1175
16.2.17 CLASS <b>ImageLoader</b> . . . . .	1176
16.2.18 CLASS <b>ImageValueActor</b> . . . . .	1177
16.2.19 CLASS <b>IndexActor</b> . . . . .	1182
16.2.20 CLASS <b>InputComponent</b> . . . . .	1187
16.2.21 CLASS <b>InputValidator</b> . . . . .	1211
16.2.22 CLASS <b>InstanceActor</b> . . . . .	1212
16.2.23 CLASS <b>LinesAndText</b> . . . . .	1217
16.2.24 CLASS <b>MessageActor</b> . . . . .	1223
16.2.25 CLASS <b>MethodStage</b> . . . . .	1228
16.2.26 CLASS <b>ObjectStage</b> . . . . .	1234
16.2.27 CLASS <b>OMIActor</b> . . . . .	1241
16.2.28 CLASS <b>OperatorActor</b> . . . . .	1246
16.2.29 CLASS <b>PanelActor</b> . . . . .	1250
16.2.30 CLASS <b>PanelController</b> . . . . .	1256
16.2.31 CLASS <b>ReferenceActor</b> . . . . .	1257
16.2.32 CLASS <b>ReferenceVariableActor</b> . . . . .	1264
16.2.33 CLASS <b>Scratch</b> . . . . .	1270
16.2.34 CLASS <b>SMIActor</b> . . . . .	1276



16.2.35 CLASS	<b>Theater</b>	1280
16.2.36 CLASS	<b>TheaterManager</b>	1307
16.2.37 CLASS	<b>ThreadController</b>	1311
16.2.38 CLASS	<b>Trace</b>	1312
16.2.39 CLASS	<b>ValueActor</b>	1313
16.2.40 CLASS	<b>VariableActor</b>	1318
16.2.41 CLASS	<b>VariableInArrayActor</b>	1323
<b>17 Package</b>	<b>koala.dynamicjava.interpreter.error</b>	<b>1329</b>
17.1	Classes	1330
17.1.1	CLASS <b>CatchedExceptionError</b>	1330
17.1.2	CLASS <b>ExecutionError</b>	1332
<b>18 Package</b>	<b>jeliot</b>	<b>1336</b>
18.1	Classes	1337
18.1.1	CLASS <b>FeatureNotImplementedException</b>	1337
18.1.2	CLASS <b>Jeliot</b>	1338
<b>19 Package</b>	<b>koala.dynamicjava.gui.resource</b>	<b>1342</b>
19.1	Interfaces	1344
19.1.1	INTERFACE <b>ActionMap</b>	1344
19.1.2	INTERFACE <b>JComponentModifier</b>	1344
19.2	Classes	1345
19.2.1	CLASS <b>ButtonFactory</b>	1345
19.2.2	CLASS <b>MenuFactory</b>	1349
19.2.3	CLASS <b>MissingListenerException</b>	1355
19.2.4	CLASS <b>ResourceFormatException</b>	1357
19.2.5	CLASS <b>ResourceManager</b>	1359
19.2.6	CLASS <b>ToolBarFactory</b>	1361
<b>20 Package</b>	<b>jeliot.io</b>	<b>1365</b>
20.1	Classes	1366
20.1.1	CLASS <b>Input</b>	1366
20.1.2	CLASS <b>Output</b>	1367
<b>21 Package</b>	<b>koala.dynamicjava.tree</b>	<b>1369</b>
21.1	Interfaces	1375
21.1.1	INTERFACE <b>ContinueTarget</b>	1375
21.1.2	INTERFACE <b>ExpressionContainer</b>	1375
21.1.3	INTERFACE <b>ExpressionStatement</b>	1376
21.1.4	INTERFACE <b>IdentifierToken</b>	1376
21.1.5	INTERFACE <b>LeftHandSide</b>	1377
21.2	Classes	1377
21.2.1	CLASS <b>AddAssignExpression</b>	1377
21.2.2	CLASS <b>AddExpression</b>	1382
21.2.3	CLASS <b>Allocation</b>	1387
21.2.4	CLASS <b>AndExpression</b>	1391
21.2.5	CLASS <b>ArrayAccess</b>	1396
21.2.6	CLASS <b>ArrayAllocation</b>	1401
21.2.7	CLASS <b>ArrayAllocation.TypeDescriptor</b>	1406
21.2.8	CLASS <b>ArrayInitializer</b>	1407

21.2.9	CLASS	<b>ArrayType</b>	1412
21.2.10	CLASS	<b>AssignExpression</b>	1417
21.2.11	CLASS	<b>BinaryExpression</b>	1421
21.2.12	CLASS	<b>BitAndAssignExpression</b>	1426
21.2.13	CLASS	<b>BitAndExpression</b>	1430
21.2.14	CLASS	<b>BitOrAssignExpression</b>	1435
21.2.15	CLASS	<b>BitOrExpression</b>	1440
21.2.16	CLASS	<b>BlockStatement</b>	1445
21.2.17	CLASS	<b>BooleanLiteral</b>	1449
21.2.18	CLASS	<b>BooleanType</b>	1454
21.2.19	CLASS	<b>BreakStatement</b>	1459
21.2.20	CLASS	<b>ByteType</b>	1463
21.2.21	CLASS	<b>CastExpression</b>	1468
21.2.22	CLASS	<b>CatchStatement</b>	1473
21.2.23	CLASS	<b>CharacterLiteral</b>	1477
21.2.24	CLASS	<b>CharType</b>	1482
21.2.25	CLASS	<b>ClassAllocation</b>	1486
21.2.26	CLASS	<b>ClassDeclaration</b>	1492
21.2.27	CLASS	<b>ClassInitializer</b>	1498
21.2.28	CLASS	<b>ComplementExpression</b>	1502
21.2.29	CLASS	<b>ConditionalExpression</b>	1506
21.2.30	CLASS	<b>ConstructorDeclaration</b>	1512
21.2.31	CLASS	<b>ConstructorInvocation</b>	1518
21.2.32	CLASS	<b>ContinueStatement</b>	1524
21.2.33	CLASS	<b>DivideAssignExpression</b>	1528
21.2.34	CLASS	<b>DivideExpression</b>	1533
21.2.35	CLASS	<b>DoStatement</b>	1538
21.2.36	CLASS	<b>DoubleLiteral</b>	1544
21.2.37	CLASS	<b>DoubleType</b>	1548
21.2.38	CLASS	<b>EmptyStatement</b>	1553
21.2.39	CLASS	<b>EqualExpression</b>	1557
21.2.40	CLASS	<b>ExclusiveOrAssignExpression</b>	1562
21.2.41	CLASS	<b>ExclusiveOrExpression</b>	1566
21.2.42	CLASS	<b>Expression</b>	1571
21.2.43	CLASS	<b>FieldAccess</b>	1575
21.2.44	CLASS	<b>FieldDeclaration</b>	1579
21.2.45	CLASS	<b>FloatLiteral</b>	1584
21.2.46	CLASS	<b>FloatType</b>	1589
21.2.47	CLASS	<b>FormalParameter</b>	1594
21.2.48	CLASS	<b>ForStatement</b>	1599
21.2.49	CLASS	<b>FunctionCall</b>	1605
21.2.50	CLASS	<b>GreaterExpression</b>	1610
21.2.51	CLASS	<b>GreaterOrEqualExpression</b>	1615
21.2.52	CLASS	<b>Identifier</b>	1620
21.2.53	CLASS	<b>IfThenElseStatement</b>	1621
21.2.54	CLASS	<b>IfThenStatement</b>	1627
21.2.55	CLASS	<b>ImportDeclaration</b>	1632
21.2.56	CLASS	<b>Initializer</b>	1637
21.2.57	CLASS	<b>InnerAllocation</b>	1641
21.2.58	CLASS	<b>InnerClassAllocation</b>	1646

21.2.59	CLASS	<b>InstanceInitializer</b>	1652
21.2.60	CLASS	<b>InstanceOfExpression</b>	1656
21.2.61	CLASS	<b>IntegerLiteral</b>	1661
21.2.62	CLASS	<b>InterfaceDeclaration</b>	1667
21.2.63	CLASS	<b>IntType</b>	1672
21.2.64	CLASS	<b>LabeledStatement</b>	1676
21.2.65	CLASS	<b>LessExpression</b>	1681
21.2.66	CLASS	<b>LessOrEqualExpression</b>	1686
21.2.67	CLASS	<b>Literal</b>	1691
21.2.68	CLASS	<b>LongLiteral</b>	1696
21.2.69	CLASS	<b>LongType</b>	1702
21.2.70	CLASS	<b>MethodCall</b>	1706
21.2.71	CLASS	<b>MethodDeclaration</b>	1711
21.2.72	CLASS	<b>MinusExpression</b>	1717
21.2.73	CLASS	<b>MultiplyAssignExpression</b>	1722
21.2.74	CLASS	<b>MultiplyExpression</b>	1727
21.2.75	CLASS	<b>Node</b>	1732
21.2.76	CLASS	<b>NotEqualExpression</b>	1736
21.2.77	CLASS	<b>NotExpression</b>	1741
21.2.78	CLASS	<b>NullLiteral</b>	1745
21.2.79	CLASS	<b>ObjectFieldAccess</b>	1750
21.2.80	CLASS	<b>ObjectMethodCall</b>	1755
21.2.81	CLASS	<b>OrExpression</b>	1760
21.2.82	CLASS	<b>PackageDeclaration</b>	1765
21.2.83	CLASS	<b>PlusExpression</b>	1770
21.2.84	CLASS	<b>PostDecrement</b>	1774
21.2.85	CLASS	<b>PostIncrement</b>	1778
21.2.86	CLASS	<b>PreDecrement</b>	1783
21.2.87	CLASS	<b>PreIncrement</b>	1787
21.2.88	CLASS	<b>PrimaryExpression</b>	1792
21.2.89	CLASS	<b>PrimitiveType</b>	1795
21.2.90	CLASS	<b>QualifiedName</b>	1800
21.2.91	CLASS	<b>ReferenceType</b>	1805
21.2.92	CLASS	<b>RemainderAssignExpression</b>	1810
21.2.93	CLASS	<b>RemainderExpression</b>	1815
21.2.94	CLASS	<b>ReturnStatement</b>	1820
21.2.95	CLASS	<b>ShiftLeftAssignExpression</b>	1824
21.2.96	CLASS	<b>ShiftLeftExpression</b>	1829
21.2.97	CLASS	<b>ShiftRightAssignExpression</b>	1834
21.2.98	CLASS	<b>ShiftRightExpression</b>	1838
21.2.99	CLASS	<b>ShortType</b>	1843
21.2.100	CLASS	<b>SimpleAllocation</b>	1847
21.2.101	CLASS	<b>SimpleAssignExpression</b>	1852
21.2.102	CLASS	<b>Statement</b>	1857
21.2.103	CLASS	<b>StaticFieldAccess</b>	1861
21.2.104	CLASS	<b>StaticMethodCall</b>	1866
21.2.105	CLASS	<b>StringLiteral</b>	1871
21.2.106	CLASS	<b>SubtractAssignExpression</b>	1876
21.2.107	CLASS	<b>SubtractExpression</b>	1881
21.2.108	CLASS	<b>SuperFieldAccess</b>	1886

21.2.10	CLASS	<b>SuperMethodCall</b>	1890
21.2.11	CLASS	<b>SwitchBlock</b>	1895
21.2.11	CLASS	<b>SwitchStatement</b>	1900
21.2.11	CLASS	<b>SynchronizedStatement</b>	1905
21.2.11	CLASS	<b>ThisExpression</b>	1910
21.2.11	CLASS	<b>ThrowStatement</b>	1914
21.2.11	CLASS	<b>TreeUtilities</b>	1919
21.2.11	CLASS	<b>TryStatement</b>	1920
21.2.11	CLASS	<b>Type</b>	1924
21.2.11	CLASS	<b>TypeDeclaration</b>	1928
21.2.11	CLASS	<b>TypeExpression</b>	1933
21.2.12	CLASS	<b>UnaryExpression</b>	1938
21.2.12	CLASS	<b>UnsignedShiftRightAssignExpression</b>	1942
21.2.12	CLASS	<b>UnsignedShiftRightExpression</b>	1947
21.2.12	CLASS	<b>VariableDeclaration</b>	1952
21.2.12	CLASS	<b>VoidType</b>	1958
21.2.12	CLASS	<b>WhileStatement</b>	1962

# Chapter 1

## Package jeliot.lang

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Classes</b>	
<b>ArrayInstance</b> .....	13
<i>The objects of this class represents an array of n-dimensions.</i>	
<b>ArrayUtilities</b> .....	16
<i>A class to help the array handling.</i>	
<b>ClassInfo</b> .....	17
<i>Object of this class contains information about a single Java class.</i>	
<b>Instance</b> .....	19
<i>The base class for all the instances.</i>	
<b>MethodFrame</b> .....	21
<i>MethodFrame is an instance of a method under execution.</i>	
<b>ObjectFrame</b> .....	23
<i>ObjectFrame represents an instance of a class that is created at run-time.</i>	
<b>Reference</b> .....	25
<i>Reference is a value of a reference type meaning that all the references to instances are objects of this class.</i>	
<b>Value</b> .....	27
<i>Value represents any primitive type of value plus String type and is the base class for reference values.</i>	
<b>Variable</b> .....	29
<i>Variable is an instance of a variable - a field or a local variable.</i>	
<b>VariableInArray</b> .....	30
<i>VariableInArray is an instance of an array variable.</i>	
<hr/>	

## 1.1 Classes

### 1.1.1 CLASS **ArrayInstance**

---

The objects of this class represents an array of n-dimensions. TODO: Maybe this should be changed to be arrays of one dimension and the chain them properly.

#### DECLARATION

---

```
public class ArrayInstance
extends jeliot.lang.Instance
```

#### FIELDS

- 
- private Object array
    - The array object.
  - private String componentType
    - The string presentation of the type of the component.
  - private int dimensions
    - The length of each dimension in the array.
  - private ArrayActor arrayActor
    - The corresponding array actor.

#### CONSTRUCTORS

- 
- *ArrayInstance*

```
public ArrayInstance( java.lang.String hashCode, java.lang.String
componentType, int [] dimensions )
```

    - **Usage**
      - \* Creates the array and sets inside the VariableInArray object which again contain the Value objects of the corresponding component type.
    - **Parameters**
      - \* **hashCode** - the hashCode of the array.
      - \* **componentType** - The component type of the array.
      - \* **dimensions** - The lengths of the dimensions of the array.

METHODS

---

- *getArrayActor*  
`public ArrayActor getArrayActor( )`
  - **Usage**
    - \* Returns the corresponding array actor.
  - **Returns** - the array actor of this array.

---
- *getComponentType*  
`public String getComponentType( )`
  - **Usage**
    - \* Returns String presentation of the component type of the array
  - **Returns** - String presentation of the component type.

---
- *getDimensionNumber*  
`public int getDimensionNumber( )`
  - **Usage**
    - \* Gives the number of dimensions in the array
  - **Returns** - The number of dimensions in the array

---
- *getDimensions*  
`public int getDimensions( )`
  - **Usage**
    - \* Gives the length of the dimensions in the array.
  - **Returns** - The length of the dimensions in the array.

---
- *getVariableAt*  
`public VariableInArray getVariableAt( int [] index )`
  - **Usage**
    - \* Returns the array variable from the given index in the array.
  - **Parameters**
    - \* **index** - an array containing the indeces for all dimensions of the array.
  - **Returns** - The VariableInArray object from the given index of the array.

---
- *length*  
`public int length( )`
  - **Usage**
    - \* The dimensions of the array. Is this needed because there is the other method doing the same?
  - **Returns** - The dimensions of the array

---
- *setArrayActor*  
`public void setArrayActor( jeliot.theater.ArrayActor aa )`
  - **Usage**

- \* Sets the corresponding ArrayActor.
  - **Parameters**
    - \* **aa** - Array actor for this array.
- 
- *setValueAt*  
 public void **setValueAt**( int [] **index**, jeliot.lang.Value **newValue** )
    - **Usage**
      - \* Assigns the given value (second parameter) into the VariableInArray in the given index (first parameter) of the array.
    - **Parameters**
      - \* **index** - The index of the array
      - \* **newValue** - The new value of the VariableInArray.

#### METHODS INHERITED FROM CLASS jeliot.lang.Instance

---

( in 1.1.4, page 19)

- *dereference*  
 public void **dereference**( )
 

---
- *getActor*  
 public InstanceActor **getActor**( )
  - **Returns -**

---
- *getHashCode*  
 public String **getHashCode**( )
  - **Returns -**

---
- *getNumberOfReferences*  
 public int **getNumberOfReferences**( )
  - **Returns -**

---
- *getType*  
 public String **getType**( )
  - **Returns -**

---
- *reference*  
 public void **reference**( )
 

---
- *setActor*  
 public void **setActor**( jeliot.theater.InstanceActor **actor** )
  - **Parameters**
    - \* **actor** -

---
- *setHashCode*  
 public void **setHashCode**( java.lang.String **hashCode** )
  - **Parameters**
    - \* **hashCode** -

---
- *setType*  
 public void **setType**( java.lang.String **type** )
  - **Parameters**
    - \* **type** -



### 1.1.2 CLASS ArrayUtilities

---

A class to help the array handling.

#### DECLARATION

---

```
public class ArrayUtilities
extends java.lang.Object
```

#### CONSTRUCTORS

---

- *ArrayUtilities*  
public **ArrayUtilities**( )

#### METHODS

---

- *getObjectAt*  
public static Object **getObjectAt**( java.lang.Object array, int [] index )  
 – **Usage**  
 \* Returns the object from the given array and from the given index.  
 – **Parameters**  
 \* **array** - The array from where the object should be returned.  
 \* **index** - The index from where the object should be returned.  
 – **Returns** - The object from the given array and given index.
- *nextIndex*  
public static boolean **nextIndex**( int [] indexCounters, int [] lengths )  
 – **Usage**  
 \* sets the given index (first parameter) to point to the next index of the array in depth first order. If the all the indeces are visited then a false is returned otherwise a true is returned.  
 – **Parameters**  
 \* **indexCounters** - The current index of the array.  
 \* **lengths** - The lenght of the dimensions in the array.  
 – **Returns** - true if there is a next index otherwise a false is returned.
- *setObjectAt*  
public static void **setObjectAt**( java.lang.Object array, int [] index, java.lang.Object newObject )  
 – **Usage**  
 \* Assigns the given object into the given array and the given index.  
 – **Parameters**  
 \* **array** - The array in which the object should be assigned.  
 \* **index** - The index in which the object should be assigned.  
 \* **newObject** - The value to be assigned

### 1.1.3 CLASS ClassInfo

---

Object of this class contains information about a single Java class. It contains all the fields, methods and constructors of the class.

#### DECLARATION

---

```
public class ClassInfo
extends java.lang.Object
```

#### FIELDS

---

- private Hashtable methods
  -
- private Hashtable fields
  -
- private Hashtable constructors
  -
- private String extendedClass
  -
- private String name
  -

#### CONSTRUCTORS

---

- *ClassInfo*

```
public ClassInfo( java.lang.Class  declaredClass )
```

  - **Parameters**
    - \* declaredClass -

---
- *ClassInfo*

```
public ClassInfo( java.lang.String  name )
```

  - **Parameters**
    - \* name -

METHODS

---

• *declareConstructor*

```
public void declareConstructor( java.lang.String key, java.lang.String
info )
```

– **Parameters**

- \* key -
  - \* info -
- 

• *declareField*

```
public void declareField( java.lang.String key, java.lang.String info )
```

– **Parameters**

- \* key -
  - \* info -
- 

• *declareMethod*

```
public void declareMethod( java.lang.String key, java.lang.String info )
```

– **Parameters**

- \* key -
  - \* info -
- 

• *extendClass*

```
public void extendClass( jeliot.lang.ClassInfo ci )
```

– **Parameters**

- \* ci -
- 

• *getConstructorInfo*

```
public String getConstructorInfo( java.lang.String key )
```

– **Parameters**

- \* key -

– **Returns** -  

---

• *getConstructors*

```
public Hashtable getConstructors( )
```

– **Returns** -  

---

• *getFieldInfo*

```
public String getFieldInfo( java.lang.String key )
```

– **Parameters**

- \* key -

– **Returns** -  

---

• *getFieldNumber*

```
public int getFieldNumber( )
```

– **Returns** -

- 
- *getFields*  
 public Hashtable **getFields**( )  
 – **Returns** -

---

  - *getMethodInfo*  
 public String **getMethodInfo**( java.lang.String key )  
 – **Parameters**  
   \* key -  
 – **Returns** -

---

  - *getMethods*  
 public Hashtable **getMethods**( )  
 – **Returns** -

---

  - *getName*  
 public String **getName**( )  
 – **Returns** -

---

  - *setDeclaredConstructors*  
 public void **setDeclaredConstructors**( java.lang.reflect.Constructor [] constructors )  
 – **Parameters**  
   \* constructors -

---

  - *setDeclaredFields*  
 public void **setDeclaredFields**( java.lang.reflect.Field [] fields )  
 – **Parameters**  
   \* fields -

---

  - *setDeclaredMethods*  
 public void **setDeclaredMethods**( java.lang.reflect.Method [] methods )  
 – **Parameters**  
   \* methods -

#### 1.1.4 CLASS Instance

---

The base class for all the instances.

##### DECLARATION

---

```
public class Instance
extends java.lang.Object
```

FIELDS

---

- private String type  
—
- private InstanceActor actor  
—
- private String hashCode  
—
- private int references  
—

CONSTRUCTORS

---

- *Instance*  
protected **Instance**( java.lang.String hashCode )  
— **Parameters**  
    \* hashCode -  
—
- *Instance*  
protected **Instance**( java.lang.String hashCode, java.lang.String type )  
— **Parameters**  
    \* hashCode -  
    \* type -

METHODS

---

- *dereference*  
public void dereference( )
- *getActor*  
public InstanceActor **getActor**( )  
— **Returns** -  
—
- *getHashCode*  
public String **getHashCode**( )  
— **Returns** -  
—
- *getNumberOfReferences*  
public int **getNumberOfReferences**( )  
— **Returns** -  
—

- *getType*  
`public String getType( )`  
 – **Returns -**  


---
- *reference*  
`public void reference( )`  


---
- *setActor*  
`public void setActor( jeliot.theater.InstanceActor actor )`  
 – **Parameters**  
   \* actor -  


---
- *setHashCode*  
`public void setHashCode( java.lang.String hashCode )`  
 – **Parameters**  
   \* hashCode -  


---
- *setType*  
`public void setType( java.lang.String type )`  
 – **Parameters**  
   \* type -

### 1.1.5 CLASS MethodFrame

---

MethodFrame is an instance of a method under execution. A method frame is created runtime each time a method is called.

#### DECLARATION

---

```
public class MethodFrame
extends java.lang.Object
```

#### FIELDS

---

- private MethodStage stage  
 –
- private Stack vars  
 –
- private int vcount  
 –
- private String name  
 –

CONSTRUCTORS

---

- *MethodFrame*  
 public **MethodFrame**( java.lang.String name )  
 – **Parameters**  
 \* name -

METHODS

---

- *closeScope*  
 public void **closeScope**( )  


---
- *declareVariable*  
 public Variable **declareVariable**( jeliot.lang.Variable var )  
 – **Parameters**  
 \* var -  
 – **Returns** -  


---
- *getMethodName*  
 public String **getMethodName**( )  
 – **Returns** -  


---
- *getMethodStage*  
 public MethodStage **getMethodStage**( )  
 – **Returns** -  


---
- *getVarCount*  
 public int **getVarCount**( )  
 – **Returns** -  


---
- *getVariable*  
 public Variable **getVariable**( java.lang.String name )  
 – **Parameters**  
 \* name -  
 – **Returns** -  


---
- *openScope*  
 public void **openScope**( )  


---
- *setMethodStage*  
 public void **setMethodStage**( jeliot.theater.MethodStage stage )  
 – **Parameters**  
 \* stage -

### 1.1.6 CLASS **ObjectFrame**

---

**ObjectFrame** represents an instance of a class that is created at run-time.

#### DECLARATION

---

```
public class ObjectFrame
extends jeliot.lang.Instance
```

#### FIELDS

---

- private **ObjectStage** stage  
—
- private **Hashtable** vars  
—
- private **int** vcount  
—
- private **String** name  
—

#### CONSTRUCTORS

---

- *ObjectFrame*  
 public **ObjectFrame**( java.lang.String hashCode, java.lang.String type,  
                   int vcount )  
   — **Parameters**  
     \* hashCode -  
     \* type -  
     \* vcount -

#### METHODS

---

- *declareVariable*  
 public **Variable** declareVariable( jeliot.lang.Variable var )  
   — **Parameters**  
     \* var -  
   — **Returns** -
- *getObjectName*  
 public **String** getObjectName( )



– **Returns** -

- *getObjectStage*

public ObjectStage getObjectStage( )

– **Returns** -

- *getVarCount*

public int getVarCount( )

– **Returns** -

- *getVariable*

public Variable getVariable( java.lang.String name )

– **Parameters**

\* name -

– **Returns** -

- *setObjectStage*

public void setObjectStage( jeliot.theater.ObjectStage stage )

– **Parameters**

\* stage -

#### METHODS INHERITED FROM CLASS jeliot.lang.Instance

---

( in 1.1.4, page 19)

- *dereference*

public void dereference( )

- *getActor*

public InstanceActor getActor( )

– **Returns** -

- *getHashCode*

public String getHashCode( )

– **Returns** -

- *getNumberOfReferences*

public int getNumberOfReferences( )

– **Returns** -

- *getType*

public String getType( )

– **Returns** -

- *reference*

public void reference( )

- *setActor*

public void setActor( jeliot.theater.InstanceActor actor )

– **Parameters**

\* actor -

- 
- *setHashCode*  
 public void **setHashCode**( java.lang.String hashCode )  
 – **Parameters**  
   \* hashCode -
  - *setType*  
 public void **setType**( java.lang.String type )  
 – **Parameters**  
   \* type -

### 1.1.7 CLASS Reference

---

Reference is a value of a reference type meaning that all the references to instances are objects of this class.

#### DECLARATION

---

```
public class Reference
extends jeliot.lang.Value
```

#### FIELDS

---

- private Instance instance  
 –
- private boolean referenced  
 –

#### CONSTRUCTORS

---

- *Reference*  
 public **Reference**( )  


---
- *Reference*  
 public **Reference**( jeliot.lang.Instance instance )  
 – **Parameters**  
   \* instance -

## METHODS

- 
- *clone*  
`public Object clone( )`


---
  - *finalize*  
`protected void finalize( )`


---
  - *getInstance*  
`public Instance getInstance( )`  

– Returns -

---
  - *makeReference*  
`public void makeReference( )`


---
  - *setInstance*  
`public void setInstance( jeliot.lang.Instance inst )`  

– Parameters

\* inst -

---
  - *setReferenced*  
`public void setReferenced( boolean value )`  

– Parameters

\* value -

---
  - *unmakeReference*  
`public void unmakeReference( )`


---

METHODS INHERITED FROM CLASS `jeliot.lang.Value`


---

( in 1.1.8, page 27)

- *clone*  
`public Object clone( )`


---
- *getActor*  
`public ValueActor getActor( )`  

– Returns -

---
- *getId*  
`public int getId( )`  

– Returns -

---
- *getType*  
`public String getType( )`  

– Returns -

---
- *getValue*  
`public String getValue( )`  

– Returns -

---

- *newBoolean*  

```
public static Value newBoolean( boolean b )
```

  - **Parameters**
    - \* b -
  - **Returns -**

---
- *setActor*  

```
public void setActor( jeliot.theater.ValueActor actor )
```

  - **Parameters**
    - \* actor -

---
- *setId*  

```
public void setId( int id )
```

  - **Parameters**
    - \* id -

---
- *toString*  

```
public String toString( )
```

### 1.1.8 CLASS Value

---

Value represents any primitive type of value plus String type and is the base class for reference values.

#### DECLARATION

---

```
public class Value
extends java.lang.Object
implements java.lang.Cloneable
```

#### FIELDS

---

- private String type
  - A String representation of the type of the value.
- private String val
  - A String describing the represented value. Can be, for example Integer, Char or if the value is actually a reference it will be the hashCode that identifies the object.
- private ValueActor actor
  - The actor of this values.
- private int id
  - The id of the value. This is used for searching a certain value.

CONSTRUCTORS

---

- *Value*  
 public Value( java.lang.String val, java.lang.String type )  
 – Parameters  
   \* val -  
   \* type -  


---
- *Value*  
 public Value( java.lang.String val, java.lang.String type, int id )  
 – Parameters  
   \* val -  
   \* type -  
   \* id -

METHODS

---

- *clone*  
 public Object clone( )  


---
- *getActor*  
 public ValueActor getActor( )  
 – Returns -  


---
- *getId*  
 public int getId( )  
 – Returns -  


---
- *getType*  
 public String getType( )  
 – Returns -  


---
- *getValue*  
 public String getValue( )  
 – Returns -  


---
- *newBoolean*  
 public static Value newBoolean( boolean b )  
 – Parameters  
   \* b -  
 – Returns -  


---
- *setActor*  
 public void setActor( jeliot.theater.ValueActor actor )  
 – Parameters

- \* actor -

---

  - *setId*  
public void setId( int id )
  - Parameters
  - \* id -

---
- *toString*  
public String toString( )

### 1.1.9 CLASS Variable

---

Variable is an instance of a variable - a field or a local variable. A new variable is created runtime every time a local variable is declared.

#### DECLARATION

---

```
public class Variable
extends java.lang.Object
```

#### CONSTRUCTORS

---

- *Variable*  
protected Variable( )
- *Variable*  
public Variable( java.lang.String name, java.lang.String type )
- Parameters
  - \* name -
  - \* type -

#### METHODS

---

- *assign*  
public void assign( jeliot.lang.Value value )
- Parameters
  - \* value -

---
- *getActor*  
public VariableActor getActor( )
- Returns -

---

- *getModifier*  
public String getModifier( )

- **Returns -**

---

- *getName*  
public String getName( )
- **Returns -**

---

- *getType*  
public String getType( )
- **Returns -**

---

- *getValue*  
public Value getValue( )
- **Returns -**

---

- *setActor*  
public void setActor( jeliot.theater.VariableActor actor )
- **Parameters**  
\* actor -

---

- *setModifier*  
protected void setModifier( java.lang.String modifier )
- **Parameters**  
\* modifier -

---

- *setType*  
protected void setType( java.lang.String type )
- **Parameters**  
\* type -

### 1.1.10 CLASS VariableInArray

---

VariableInArray is an instance of an array variable. A new array variables are created at runtime every time a new array is created.

#### DECLARATION

---

```
public class VariableInArray
extends jeliot.lang.Variable
```

CONSTRUCTORS

---

- *VariableInArray*  
public VariableInArray( )
- *VariableInArray*  
public VariableInArray( jeliot.lang.ArrayInstance array, java.lang.String  
componentType )
  - Parameters
    - \* array -
    - \* componentType -

METHODS INHERITED FROM CLASS jeliot.lang.Variable

---

( in 1.1.9, page 29)

- *assign*  
public void assign( jeliot.lang.Value value )
  - Parameters
    - \* value -
- *getActor*  
public VariableActor getActor( )
  - Returns -
- *getModifier*  
public String getModifier( )
  - Returns -
- *getName*  
public String getName( )
  - Returns -
- *getType*  
public String getType( )
  - Returns -
- *getValue*  
public Value getValue( )
  - Returns -
- *setActor*  
public void setActor( jeliot.theater.VariableActor actor )
  - Parameters
    - \* actor -
- *setModifier*  
protected void setModifier( java.lang.String modifier )
  - Parameters
    - \* modifier -
- *setType*  
protected void setType( java.lang.String type )
  - Parameters
    - \* type -



## Chapter 2

# Package koala.dynamicjava.parser

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Interfaces</b>	
<b>Parser.CompileTimeExpression</b> .....	34
<i>...no description...</i>	
<b>ParserConstants</b> .....	34
<i>...no description...</i>	
<b>Classes</b>	
<b>ASCII_UCodeESC_CharStream</b> .....	41
<i>An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (with java-like unicode escape processing).</i>	
<b>ParseException</b> .....	44
<i>This exception is thrown when parse errors are encountered.</i>	
<b>Parser</b> .....	47
<i>This class represents a (interpreted) Java 1.1 language parser</i>	
<b>Parser.ArgumentsSuffix</b> .....	106
<i>...no description...</i>	
<b>Parser.ArrayReferenceSuffix</b> .....	107
<i>...no description...</i>	
<b>Parser.ClassBody</b> .....	108
<i>To store a class body</i>	
<b>Parser.CompileTimeQualifiedName</b> .....	108
<i>...no description...</i>	
<b>Parser.CompileTimeSuperExpression</b> .....	113
<i>...no description...</i>	
<b>Parser.DotAllocationSuffix</b> .....	117
<i>...no description...</i>	
<b>Parser.DotIdentifierSuffix</b> .....	118
<i>...no description...</i>	
<b>Parser.DotThisSuffix</b> .....	119
<i>...no description...</i>	
<b>Parser.ExpressionSuffix</b> .....	120
<i>...no description...</i>	
<b>Parser.JJCalls</b> .....	121
<i>...no description...</i>	
<b>Parser.ModifierFlags</b> .....	121
<i>To store modifier flags</i>	

<b>ParserTokenManager</b> .....	121
<i>...no description...</i>	
<b>Token</b> .....	124
<i>Describes the input token stream.</i>	
<b>TokenMgrError</b> .....	126
<i>...no description...</i>	

---

## 2.1 Interfaces

### 2.1.1 INTERFACE **Parser.CompileTimeExpression**

---

#### DECLARATION

---

```
static interface Parser.CompileTimeExpression
```

#### METHODS

---

- *createRuntimeExpression*  

```
public Expression createRuntimeExpression( )
```

  - **Usage**
    - \* Builds a runtime expression from this compile-time one. This method must only be called when the expression has no suffix.

### 2.1.2 INTERFACE **ParserConstants**

---

#### DECLARATION

---

```
public interface ParserConstants
```

#### FIELDS

---

- public static final int EOF
  -
- public static final int SINGLE\_LINE\_COMMENT
  -
- public static final int FORMAL\_COMMENT
  -
- public static final int MULTILINE\_COMMENT
  -
- public static final int ABSTRACT
  -
- public static final int BOOLEAN
  -

- public static final int BREAK  
—
- public static final int BYTE  
—
- public static final int CASE  
—
- public static final int CATCH  
—
- public static final int CHAR  
—
- public static final int CLASS  
—
- public static final int CONST  
—
- public static final int CONTINUE  
—
- public static final int \_DEFAULT  
—
- public static final int DO  
—
- public static final int DOUBLE  
—
- public static final int ELSE  
—
- public static final int EXTENDS  
—
- public static final int FALSE  
—
- public static final int FINAL  
—
- public static final int FINALLY  
—
- public static final int FLOAT  
—

- public static final int FOR  
—
- public static final int GOTO  
—
- public static final int IF  
—
- public static final int IMPLEMENTS  
—
- public static final int IMPORT  
—
- public static final int INSTANCEOF  
—
- public static final int INT  
—
- public static final int INTERFACE  
—
- public static final int LONG  
—
- public static final int NATIVE  
—
- public static final int NEW  
—
- public static final int NULL  
—
- public static final int PACKAGE  
—
- public static final int PRIVATE  
—
- public static final int PROTECTED  
—
- public static final int PUBLIC  
—
- public static final int RETURN  
—

- public static final int SHORT  
—
- public static final int STATIC  
—
- public static final int SUPER  
—
- public static final int SWITCH  
—
- public static final int SYNCHRONIZED  
—
- public static final int THIS  
—
- public static final int THROW  
—
- public static final int THROWS  
—
- public static final int TRANSIENT  
—
- public static final int TRUE  
—
- public static final int TRY  
—
- public static final int VOID  
—
- public static final int VOLATILE  
—
- public static final int WHILE  
—
- public static final int INTEGER\_LITERAL  
—
- public static final int LONG\_LITERAL  
—
- public static final int DECIMAL\_LITERAL  
—

- public static final int HEX\_LITERAL  
—
- public static final int OCTAL\_LITERAL  
—
- public static final int FLOAT\_LITERAL  
—
- public static final int DOUBLE\_LITERAL  
—
- public static final int EXPONENT  
—
- public static final int CHARACTER\_LITERAL  
—
- public static final int STRING\_LITERAL  
—
- public static final int IDENTIFIER  
—
- public static final int LETTER  
—
- public static final int DIGIT  
—
- public static final int LPAREN  
—
- public static final int RPAREN  
—
- public static final int LBRACE  
—
- public static final int RBRACE  
—
- public static final int LBRACKET  
—
- public static final int RBRACKET  
—
- public static final int SEMICOLON  
—

- public static final int COMMA  
—
- public static final int DOT  
—
- public static final int ASSIGN  
—
- public static final int GREATER\_THAN  
—
- public static final int LESS  
—
- public static final int BANG  
—
- public static final int TILDE  
—
- public static final int HOOK  
—
- public static final int COLON  
—
- public static final int EQUAL  
—
- public static final int LESS\_OR\_EQUAL  
—
- public static final int GREATER\_OR\_EQUAL  
—
- public static final int NOT\_EQUAL  
—
- public static final int CONDITIONAL\_OR  
—
- public static final int CONDITIONAL\_AND  
—
- public static final int INCREMENT  
—
- public static final int DECREMENT  
—



- public static final int PLUS  
—
- public static final int MINUS  
—
- public static final int STAR  
—
- public static final int SLASH  
—
- public static final int BITWISE\_AND  
—
- public static final int BITWISE\_OR  
—
- public static final int XOR  
—
- public static final int REMAINDER  
—
- public static final int LEFT\_SHIFT  
—
- public static final int RIGHT\_SIGNED\_SHIFT  
—
- public static final int RIGHT\_UNSIGNED\_SHIFT  
—
- public static final int PLUS\_ASSIGN  
—
- public static final int MINUS\_ASSIGN  
—
- public static final int STAR\_ASSIGN  
—
- public static final int SLASH\_ASSIGN  
—
- public static final int AND\_ASSIGN  
—
- public static final int OR\_ASSIGN  
—

- public static final int XOR\_ASSIGN  
—
- public static final int REMAINDER\_ASSIGN  
—
- public static final int LEFT\_SHIFT\_ASSIGN  
—
- public static final int RIGHT\_SIGNED\_SHIFT\_ASSIGN  
—
- public static final int RIGHT\_UNSIGNED\_SHIFT\_ASSIGN  
—
- public static final int DEFAULT  
—
- public static final int IN\_SINGLE\_LINE\_COMMENT  
—
- public static final int IN\_FORMAL\_COMMENT  
—
- public static final int IN\_MULTILINE\_COMMENT  
—
- public static final String tokenImage  
—

## 2.2 Classes

### 2.2.1 CLASS ASCII\_UCodeESC\_CharStream

---

An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (with java-like unicode escape processing).

#### DECLARATION

---

```
public final class ASCII_UCodeESC_CharStream
extends java.lang.Object
```

FIELDS

---

- public static final boolean staticFlag

—

- public int bufpos

—

- private int bufline

—

- private int bufcolumn

—

- private int column

—

- private int line

—

- private Reader inputStream

—

- private boolean prevCharIsCR

—

- private boolean prevCharIsLF

—

- private char nextCharBuf

—

- private char buffer

—

- private int maxNextCharInd

—

- private int nextCharInd

—

- private int inBuf

—

## CONSTRUCTORS

---

- *ASCII\_UCodeESC\_CharStream*  
`public ASCII_UCodeESC_CharStream( java.io.InputStream dstream, int startline, int startcolumn )`
- *ASCII\_UCodeESC\_CharStream*  
`public ASCII_UCodeESC_CharStream( java.io.InputStream dstream, int startline, int startcolumn, int buffersize )`
- *ASCII\_UCodeESC\_CharStream*  
`public ASCII_UCodeESC_CharStream( java.io.Reader dstream, int startline, int startcolumn )`
- *ASCII\_UCodeESC\_CharStream*  
`public ASCII_UCodeESC_CharStream( java.io.Reader dstream, int startline, int startcolumn, int buffersize )`

## METHODS

---

- *adjustBeginLineColumn*  
`public void adjustBeginLineColumn( int newLine, int newCol )`
  - **Usage**
    - \* Method to adjust line and column numbers for the start of a token.

- 
- *AdjustBuffSize*  
`private final void AdjustBuffSize( )`
  - *backup*  
`public final void backup( int amount )`
  - *BeginToken*  
`public final char BeginToken( )`
  - *Done*  
`public void Done( )`
  - *ExpandBuff*  
`private final void ExpandBuff( boolean wrapAround )`
  - *FillBuff*  
`private final void FillBuff( )`
  - *getBeginColumn*  
`public final int getBeginColumn( )`
  - *getBeginLine*  
`public final int getBeginLine( )`
  - *getColumn*  
`public final int getColumn( )`

– **See Also**

\* koala.dynamicjava.parser.ASCII\_UCodeESC\_CharStream.getEndColumn ( in 2.2.1, page 44)

---

- *getEndColumn*  
public final int **getEndColumn**( )

---

- *getEndLine*  
public final int **getEndLine**( )

---

- *GetImage*  
public final String **GetImage**( )

---

- *getLine*  
public final int **getLine**( )

– See Also

\* koala.dynamicjava.parser.ASCII\_UCodeESC\_CharStream.getEndLine ( in 2.2.1, page 44)

---

- *GetSuffix*  
public final char **GetSuffix**( int len )

---

- *herval*  
static final int **hexval**( char c )

---

- *ReadByte*  
private final char **ReadByte**( )

---

- *readChar*  
public final char **readChar**( )

---

- *ReInit*  
public void **ReInit**( java.io.InputStream dstream, int startline, int startcolumn )

---

- *ReInit*  
public void **ReInit**( java.io.InputStream dstream, int startline, int startcolumn, int buffersize )

---

- *ReInit*  
public void **ReInit**( java.io.Reader dstream, int startline, int startcolumn )

---

- *ReInit*  
public void **ReInit**( java.io.Reader dstream, int startline, int startcolumn, int buffersize )

---

- *UpdateLineColumn*  
private final void **UpdateLineColumn**( char c )

## 2.2.2 CLASS *ParseException*

---

This exception is thrown when parse errors are encountered. You can explicitly create objects of this exception type by calling the method `generateParseException` in the generated parser.

You can modify this class to customize your error reporting mechanisms so long as you retain the public fields.

DECLARATION

---

```
public class ParseException
extends java.lang.Exception
```

SERIALIZABLE FIELDS

---

- public Token currentToken
  - This is the last token that has been consumed successfully. If this object has been created due to a parse error, the token following this token will (therefore) be the first error token.
- public int expectedTokenSequences
  - Each entry in this array is an array of integers. Each array of integers represents a sequence of tokens (by their ordinal values) that is expected at this point of the parse.
- public String tokenImage
  - This is a reference to the "tokenImage" array of the generated parser within which the parse error occurred. This array is defined in the generated ...Constants interface.

FIELDS

---

- public Token currentToken
  - This is the last token that has been consumed successfully. If this object has been created due to a parse error, the token following this token will (therefore) be the first error token.
- public int expectedTokenSequences
  - Each entry in this array is an array of integers. Each array of integers represents a sequence of tokens (by their ordinal values) that is expected at this point of the parse.
- public String tokenImage
  - This is a reference to the "tokenImage" array of the generated parser within which the parse error occurred. This array is defined in the generated ...Constants interface.

CONSTRUCTORS

---

- *ParseException*  
 public **ParseException**( )
  - **Usage**
    - \* The following constructors are for use by you for whatever purpose you can think of. Constructing the exception in this manner makes the exception behave in the normal way - i.e., as documented in the class "Throwable". The fields "errorToken", "expectedTokenSequences", and "tokenImage" do not contain relevant information. The JavaCC generated code does not use these constructors.

---

- *ParseException*

public ParseException( java.lang.String message )

- *ParseException*

public ParseException( koala.dynamicjava.parser.Token currentTokenVal, int  
[] [] expectedTokenSequencesVal, java.lang.String [] tokenImageVal )

- Usage

- \* This constructor is used by the method "generateParseException" in the generated parser. Calling this constructor generates a new object of this type with the fields "currentToken", "expectedTokenSequences", and "tokenImage" set. The boolean flag "specialConstructor" is also set to true to indicate that this constructor was used to create this object. This constructor calls its super class with the empty string to force the "toString" method of parent class "Throwable" to print the error message in the form: ParseException: <result of getMessage>

## METHODS

---

- *add\_escapes*

protected String add\_escapes( java.lang.String str )

- Usage

- \* Used to convert raw characters to their escaped version when these raw version cannot be used as part of an ASCII string literal.

---

- *getMessage*

public String getMessage( )

- Usage

- \* This method has the standard behavior when this object has been created using the standard constructors. Otherwise, it uses "currentToken" and "expectedTokenSequences" to generate a parse error message and returns it. If this object has been created due to a parse error, and you do not catch it (it gets thrown from the parser), then this method is called during the printing of the final stack trace, and hence the correct error message gets displayed.

## METHODS INHERITED FROM CLASS java.lang.Exception

---

## METHODS INHERITED FROM CLASS java.lang.Throwable

---

- *fillInStackTrace*

public synchronized native Throwable fillInStackTrace( )

- *getCause*

public Throwable getCause( )

- *getLocalizedMessage*

public String getLocalizedMessage( )

- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

### 2.2.3 CLASS Parser

---

This class represents a (interpreted) Java 1.1 language parser

#### DECLARATION

---

```
public class Parser
extends java.lang.Object
implements ParserConstants
```



FIELDS

---

- private String filename
  - The name of the file currenty interpreted
- private LocalizedMessageReader reader
  - The message reader
- public ParserTokenManager token\_source
  -
- public Token token
  -
- public Token jj\_nt
  -
- private int jj\_ntk
  -
- private Token jj\_scanpos
  -
- private Token jj\_lastpos
  -
- private int jj\_la
  -
- public boolean lookingAhead
  -
- private boolean jj\_semLA
  -
- private int jj\_gen
  -
- private final int jj\_la1
  -
- private final int jj\_la1\_0
  -
- private final int jj\_la1\_1
  -
- private final int jj\_la1\_2
  -

- private final int jj\_la1\_3  
—
- private final Parser.JJCalls jj\_2\_rtns  
—
- private boolean jj\_rescan  
—
- private int jj\_gc  
—
- private Vector jj\_expentries  
—
- private int jj\_expentry  
—
- private int jj\_kind  
—
- private int jj\_lasttokens  
—
- private int jj\_endpos  
—

## CONSTRUCTORS

---

- *Parser*  
public Parser( java.io.InputStream stream )
- *Parser*  
public Parser( koala.dynamicjava.parser.ParserTokenManager tm )
- *Parser*  
public Parser( java.io.Reader stream )

## METHODS

---

- *additiveExpression*  
public final Expression **additiveExpression( )**  
— **Usage**  
\* Used internally to parse an expression  

---
- *additiveExpressionLookahead*  
public final void **additiveExpressionLookahead( )**  
— **Usage**

---

\* Used internally for lookahead

---

- *allocationExpression*

**public final Expression allocationExpression( )**

- **Usage**

- \* Used internally to parse an expression

---

- *allocationExpressionLookahead*

**public final void allocationExpressionLookahead( )**

- **Usage**

- \* Used internally for lookahead

---

- *andExpression*

**public final Expression andExpression( )**

- **Usage**

- \* Used internally to parse an expression

---

- *andExpressionLookahead*

**public final void andExpressionLookahead( )**

- **Usage**

- \* Used internally for lookahead

---

- *argumentList*

**public final List argumentList( )**

- **Usage**

- \* Used internally to parse an expression

---

- *arguments*

**public final Parser.ArgumentsSuffix arguments( )**

- **Usage**

- \* Used internally to parse an expression

---

- *argumentsLookahead*

**public final void argumentsLookahead( )**

- **Usage**

- \* Used internally for lookahead

---

- *arrayDimsAndInits*

**public final ArrayAllocation.TypeDescriptor arrayDimsAndInits( )**

- **Usage**

- \* Used internally to parse an expression

---

- *arrayDimsAndInitsLookahead*

**public final void arrayDimsAndInitsLookahead( )**

- **Usage**

---

\* Used internally for lookahead

---

- *arrayInitializer*

**public final ArrayInitializer arrayInitializer( )**

- **Usage**

- \* Parses an array initializer

- **See Also**

- \* `koala.dynamicjava.tree.ArrayInitializer` ( in 21.2.8, page 1407)

---

- *arrayInitializerLookahead*

**public final void arrayInitializerLookahead( )**

- **Usage**

- \* Used internally for lookahead

---

- *arrayReference*

**public final Parser.ExpressionSuffix arrayReference( )**

- **Usage**

- \* Used internally to parse an expression

---

- *assignmentOperatorLookahead*

**public final void assignmentOperatorLookahead( )**

- **Usage**

- \* Used internally for lookahead

---

- *block*

**public final BlockStatement block( )**

- **Usage**

- \* Parses a block

- **See Also**

- \* `koala.dynamicjava.tree.BlockStatement` ( in 21.2.16, page 1445)

---

- *blockLookahead*

**public final void blockLookahead( )**

- **Usage**

- \* Used internally for lookahead

---

- *blockStatement*

**public final List blockStatement( )**

- **Usage**

- \* Parses one block statement.

- **Returns** - a list of node because one variable declaration can contain multiple declarations.

- **See Also**

- \* `koala.dynamicjava.tree.Node` ( in 21.2.75, page 1732)

---

- *blockStatementLookahead*  
 public final void **blockStatementLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *breakStatement*  
 public final BreakStatement **breakStatement**( )  
 – **Usage**  
   \* Parses a break statement  
 – **See Also**  
   \* koala.dynamicjava.tree.BreakStatement ( in 21.2.19, page 1459)  


---
- *breakStatementLookahead*  
 public final void **breakStatementLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *castExpression*  
 public final Expression **castExpression**( )  
 – **Usage**  
   \* Used internally to parse an expression  


---
- *castExpressionLookahead*  
 public final void **castExpressionLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *castLookahead*  
 public final void **castLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *classBody*  
 public final Parser.ClassBody **classBody**( )  
 – **Usage**  
   \* Used internally to parse the body of a class  


---
- *classBodyDeclaration*  
 public final List **classBodyDeclaration**( )  
 – **Usage**  
   \* Parses one declaration in the body of a class.  
 – **Returns** - a list of node because one field declaration can contain multiple declarations.  
 – **See Also**

---

\* koala.dynamicjava.tree.Node ( in 21.2.75, page 1732)

---

- *classBodyDeclarationLookahead*

public final void **classBodyDeclarationLookahead**( )

- **Usage**

\* Used internally for lookahead

---

- *classBodyLookahead*

public final void **classBodyLookahead**( )

- **Usage**

\* Used internally for lookahead

---

- *classDeclaration*

public final ClassDeclaration **classDeclaration**( )

- **Usage**

\* Parses a class declaration

- **See Also**

\* koala.dynamicjava.tree.ClassDeclaration ( in 21.2.26, page 1492)

---

- *classDeclarationLookahead*

public final void **classDeclarationLookahead**( )

- **Usage**

\* Used internally for lookahead

---

- *classDeclarationLookahead2*

public final void **classDeclarationLookahead2**( )

- **Usage**

\* Used internally for lookahead

---

- *classExpression*

public final Expression **classExpression**( )

- **Usage**

\* Used internally to parse an expression

---

- *conditionalAndExpression*

public final Expression **conditionalAndExpression**( )

- **Usage**

\* Used internally to parse an expression

---

- *conditionalAndExpressionLookahead*

public final void **conditionalAndExpressionLookahead**( )

- **Usage**

\* Used internally for lookahead

---

- *conditionalExpression*  
 public final Expression **conditionalExpression**( )  
 – **Usage**  
   \* Used internally to parse an expression  


---
- *conditionalExpressionLookahead*  
 public final void **conditionalExpressionLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *conditionalOrExpression*  
 public final Expression **conditionalOrExpression**( )  
 – **Usage**  
   \* Used internally to parse an expression  


---
- *conditionalOrExpressionLookahead*  
 public final void **conditionalOrExpressionLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *constructorDeclaration*  
 public final ConstructorDeclaration **constructorDeclaration**( )  
 – **Usage**  
   \* Parses a constructor declaration  
 – **See Also**  
   \* koala.dynamicjava.tree.ConstructorDeclaration ( in 21.2.30, page 1512)  


---
- *constructorDeclarationLookahead*  
 public final void **constructorDeclarationLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *constructorDeclarationLookahead2*  
 public final void **constructorDeclarationLookahead2**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *continueStatement*  
 public final ContinueStatement **continueStatement**( )  
 – **Usage**  
   \* Parses a continue statement  
 – **See Also**  
   \* koala.dynamicjava.tree.ContinueStatement ( in 21.2.32, page 1524)  


---

- *continueStatementLookahead*  
`public final void continueStatementLookahead( )`  
  - **Usage**  
    - \* Used internally for lookahead

---
- *createFieldDeclaration*  
`private FieldDeclaration createFieldDeclaration(  
koala.dynamicjava.parser.Parser.ModifierFlags mf,  
koala.dynamicjava.tree.Type typ, koala.dynamicjava.parser.Token name,  
koala.dynamicjava.tree.Expression exp, int dim )`  
  - **Usage**  
    - \* Creates a field declaration

---
- *createPrimaryExpression*  
`private static Expression createPrimaryExpression(  
koala.dynamicjava.tree.Expression prefix, java.util.List suffixes )`  
  - **Usage**  
    - \* Creates a primary expression from a prefix and a list of suffixes
  - **Parameters**  
    - \* **prefix** - the prefix expression
    - \* **suffixes** - the suffix list built in reverse order

---
- *createVariableDeclaration*  
`private VariableDeclaration createVariableDeclaration(  
koala.dynamicjava.parser.Token mf, koala.dynamicjava.tree.Type typ,  
koala.dynamicjava.parser.Token name, koala.dynamicjava.tree.Expression  
exp, int dim )`  
  - **Usage**  
    - \* Creates a variable declaration

---
- *disable\_tracing*  
`public final void disable_tracing( )`  


---
- *doStatement*  
`public final DoStatement doStatement( )`  
  - **Usage**  
    - \* Parses a do statement
  - **See Also**  
    - \* `koala.dynamicjava.tree.DoStatement` ( in 21.2.35, page 1538)

---
- *doStatementLookahead*  
`public final void doStatementLookahead( )`  
  - **Usage**  
    - \* Used internally for lookahead

---
- *dotAllocationExpression*  
`public final Parser.ExpressionSuffix dotAllocationExpression( )`



- **Usage**
    - \* Used internally to parse an expression

---
- *dotIdentifier*

```
public final Parser.ExpressionSuffix dotIdentifier( )
```

    - **Usage**
      - \* Used internally to parse an expression

---
  - *dotThis*

```
public final Parser.ExpressionSuffix dotThis( )
```

    - **Usage**
      - \* Used internally to parse an expression

---
  - *emptyStatement*

```
public final EmptyStatement emptyStatement( )
```

      - **Usage**
        - \* Parses an empty statement
      - **See Also**
        - \* koala.dynamicjava.tree.EmptyStatement ( in 21.2.38, page 1553)

---
    - *enable\_tracing*

```
public final void enable_tracing( )
```

---
    - *equalityExpression*

```
public final Expression equalityExpression( )
```

        - **Usage**
          - \* Used internally to parse an expression

---
      - *equalityExpressionLookahead*

```
public final void equalityExpressionLookahead( )
```

        - **Usage**
          - \* Used internally for lookahead

---
      - *exclusiveOrExpression*

```
public final Expression exclusiveOrExpression( )
```

          - **Usage**
            - \* Used internally to parse an expression

---
        - *exclusiveOrExpressionLookahead*

```
public final void exclusiveOrExpressionLookahead( )
```

          - **Usage**
            - \* Used internally for lookahead

---
        - *explicitConstructorInvocation*

```
public final ConstructorInvocation explicitConstructorInvocation( )
```

- **Usage**
    - \* Parses a constructor invocation
  - **See Also**
    - \* `koala.dynamicjava.tree.ConstructorInvocation` ( in 21.2.31, page 1518)

---
- *explicitConstructorInvocationLookahead*  
**public final void explicitConstructorInvocationLookahead( )**
  - **Usage**
    - \* Used internally for lookahead

---
- *expression*  
**public final Expression expression( )**
  - **Usage**
    - \* Parses an expression
  - **See Also**
    - \* `koala.dynamicjava.tree.Expression` ( in 21.2.42, page 1571)

---
- *expressionLookahead*  
**public final void expressionLookahead( )**
  - **Usage**
    - \* Used internally for lookahead

---
- *expressionSuffix*  
**public final Expression expressionSuffix( koala.dynamicjava.tree.Expression pre )**
  - **Usage**
    - \* Used internally to parse an expression

---
- *fieldDeclaration*  
**public final List fieldDeclaration( )**
  - **Usage**
    - \* Parses a field declaration.
  - **Returns** - a list of field declaration because one field declaration can contain multiple declarations.
  - **See Also**
    - \* `koala.dynamicjava.tree.FieldDeclaration` ( in 21.2.44, page 1579)

---
- *fieldDeclarationLookahead*  
**public final void fieldDeclarationLookahead( )**
  - **Usage**
    - \* Used internally for lookahead

---
- *forInit*  
**public final List forInit( )**
  - **Usage**

\* Used internally to parse a 'for' statement

---

- *forInitLookahead*

**public final void forInitLookahead( )**

- **Usage**

\* Used internally for lookahead

---

- *formalParameter*

**public final FormalParameter formalParameter( )**

- **Usage**

\* Parses one formal parameter

- **See Also**

\* `koala.dynamicjava.tree.FormalParameter` ( in 21.2.47, page 1594)

---

- *formalParameterLookahead*

**public final void formalParameterLookahead( )**

- **Usage**

\* Used internally for lookahead

---

- *formalParameters*

**public final List formalParameters( )**

- **Usage**

\* Parses formal parameters of the form '(param, param, ...)'

- **See Also**

\* `koala.dynamicjava.tree.FormalParameter` ( in 21.2.47, page 1594)

---

- *formalParametersLookahead*

**public final void formalParametersLookahead( )**

- **Usage**

\* Used internally for lookahead

---

- *forStatement*

**public final ForStatement forStatement( )**

- **Usage**

\* Parses a for statement

- **See Also**

\* `koala.dynamicjava.tree.ForStatement` ( in 21.2.48, page 1599)

---

- *forStatementLookahead*

**public final void forStatementLookahead( )**

- **Usage**

\* Used internally for lookahead

---

- *generateParseException*

**public final ParseException generateParseException( )**

---

- *getNextToken*  


---

**public final Token getNextToken( )**
- *getToken*  


---

**public final Token getToken( int index )**
- *ifStatement*  
**public final Statement ifStatement( )**
  - **Usage**
    - \* Parses an if statement
  - **See Also**
    - \* `koala.dynamicjava.tree.IfThenStatement` ( in 21.2.54, page 1627)
    - \* `koala.dynamicjava.tree.IfThenElseStatement` ( in 21.2.53, page 1621)

---
- *ifStatementLookahead*  
**public final void ifStatementLookahead( )**
  - **Usage**
    - \* Used internally for lookahead

---
- *importDeclaration*  
**public final ImportDeclaration importDeclaration( )**
  - **Usage**
    - \* Parses an import declaration
  - **See Also**
    - \* `koala.dynamicjava.tree.ImportDeclaration` ( in 21.2.55, page 1632)

---
- *inclusiveOrExpression*  
**public final Expression inclusiveOrExpression( )**
  - **Usage**
    - \* Used internally to parse an expression

---
- *inclusiveOrExpressionLookahead*  
**public final void inclusiveOrExpressionLookahead( )**
  - **Usage**
    - \* Used internally for lookahead

---
- *initializer*  
**public final Initializer initializer( )**
  - **Usage**
    - \* Parses one initializer
  - **See Also**
    - \* `koala.dynamicjava.tree.Initializer` ( in 21.2.56, page 1637)

---
- *initializerLookahead*  
**public final void initializerLookahead( )**
  - **Usage**

- - \* Used internally for lookahead
- - *instanceOfExpression*  
 public final Expression **instanceOfExpression**( )
    - **Usage**
      - \* Used internally to parse an expression
- - *instanceOfExpressionLookahead*  
 public final void **instanceOfExpressionLookahead**( )
    - **Usage**
      - \* Used internally for lookahead
- - *interfaceDeclaration*  
 public final InterfaceDeclaration **interfaceDeclaration**( )
    - **Usage**
      - \* Parses a interface declaration
    - **See Also**
      - \* koala.dynamicjava.tree.InterfaceDeclaration ( in 21.2.62, page 1667)
- - *interfaceDeclarationLookahead*  
 public final void **interfaceDeclarationLookahead**( )
    - **Usage**
      - \* Used internally for lookahead
- - *interfaceDeclarationLookahead2*  
 public final void **interfaceDeclarationLookahead2**( )
    - **Usage**
      - \* Used internally for lookahead
- - *interfaceMemberDeclaration*  
 public final List **interfaceMemberDeclaration**( )
    - **Usage**
      - \* Parses one declaration in the body of an interface.
    - **Returns** - a list of node because one field declaration can contain multiple declarations.
    - **See Also**
      - \* koala.dynamicjava.tree.Node ( in 21.2.75, page 1732)
- - *interfaceMemberDeclarationLookahead*  
 public final void **interfaceMemberDeclarationLookahead**( )
    - **Usage**
      - \* Used internally for lookahead
- - *jj\_2\_1*  
 private final boolean **jj\_2\_1**( int xla )

- *jj\_2\_10*  
private final boolean **jj\_2\_10**( int xla )
- *jj\_2\_11*  
private final boolean **jj\_2\_11**( int xla )
- *jj\_2\_12*  
private final boolean **jj\_2\_12**( int xla )
- *jj\_2\_13*  
private final boolean **jj\_2\_13**( int xla )
- *jj\_2\_14*  
private final boolean **jj\_2\_14**( int xla )
- *jj\_2\_15*  
private final boolean **jj\_2\_15**( int xla )
- *jj\_2\_16*  
private final boolean **jj\_2\_16**( int xla )
- *jj\_2\_17*  
private final boolean **jj\_2\_17**( int xla )
- *jj\_2\_18*  
private final boolean **jj\_2\_18**( int xla )
- *jj\_2\_19*  
private final boolean **jj\_2\_19**( int xla )
- *jj\_2\_2*  
private final boolean **jj\_2\_2**( int xla )
- *jj\_2\_20*  
private final boolean **jj\_2\_20**( int xla )
- *jj\_2\_21*  
private final boolean **jj\_2\_21**( int xla )
- *jj\_2\_22*  
private final boolean **jj\_2\_22**( int xla )
- *jj\_2\_23*  
private final boolean **jj\_2\_23**( int xla )
- *jj\_2\_24*  
private final boolean **jj\_2\_24**( int xla )
- *jj\_2\_25*  
private final boolean **jj\_2\_25**( int xla )
- *jj\_2\_26*  
private final boolean **jj\_2\_26**( int xla )
- *jj\_2\_27*  
private final boolean **jj\_2\_27**( int xla )
- *jj\_2\_28*  
private final boolean **jj\_2\_28**( int xla )

- *jj\_2\_29*  
private final boolean **jj\_2\_29**( int xla )
- *jj\_2\_3*  
private final boolean **jj\_2\_3**( int xla )
- *jj\_2\_30*  
private final boolean **jj\_2\_30**( int xla )
- *jj\_2\_31*  
private final boolean **jj\_2\_31**( int xla )
- *jj\_2\_32*  
private final boolean **jj\_2\_32**( int xla )
- *jj\_2\_33*  
private final boolean **jj\_2\_33**( int xla )
- *jj\_2\_34*  
private final boolean **jj\_2\_34**( int xla )
- *jj\_2\_35*  
private final boolean **jj\_2\_35**( int xla )
- *jj\_2\_36*  
private final boolean **jj\_2\_36**( int xla )
- *jj\_2\_37*  
private final boolean **jj\_2\_37**( int xla )
- *jj\_2\_38*  
private final boolean **jj\_2\_38**( int xla )
- *jj\_2\_39*  
private final boolean **jj\_2\_39**( int xla )
- *jj\_2\_4*  
private final boolean **jj\_2\_4**( int xla )
- *jj\_2\_40*  
private final boolean **jj\_2\_40**( int xla )
- *jj\_2\_41*  
private final boolean **jj\_2\_41**( int xla )
- *jj\_2\_42*  
private final boolean **jj\_2\_42**( int xla )
- *jj\_2\_43*  
private final boolean **jj\_2\_43**( int xla )
- *jj\_2\_44*  
private final boolean **jj\_2\_44**( int xla )
- *jj\_2\_45*  
private final boolean **jj\_2\_45**( int xla )
- *jj\_2\_46*  
private final boolean **jj\_2\_46**( int xla )

- *jj\_2\_47*  
private final boolean **jj\_2\_47**( int xla )
- *jj\_2\_48*  
private final boolean **jj\_2\_48**( int xla )
- *jj\_2\_49*  
private final boolean **jj\_2\_49**( int xla )
- *jj\_2\_5*  
private final boolean **jj\_2\_5**( int xla )
- *jj\_2\_50*  
private final boolean **jj\_2\_50**( int xla )
- *jj\_2\_51*  
private final boolean **jj\_2\_51**( int xla )
- *jj\_2\_52*  
private final boolean **jj\_2\_52**( int xla )
- *jj\_2\_53*  
private final boolean **jj\_2\_53**( int xla )
- *jj\_2\_54*  
private final boolean **jj\_2\_54**( int xla )
- *jj\_2\_55*  
private final boolean **jj\_2\_55**( int xla )
- *jj\_2\_56*  
private final boolean **jj\_2\_56**( int xla )
- *jj\_2\_57*  
private final boolean **jj\_2\_57**( int xla )
- *jj\_2\_58*  
private final boolean **jj\_2\_58**( int xla )
- *jj\_2\_59*  
private final boolean **jj\_2\_59**( int xla )
- *jj\_2\_6*  
private final boolean **jj\_2\_6**( int xla )
- *jj\_2\_7*  
private final boolean **jj\_2\_7**( int xla )
- *jj\_2\_8*  
private final boolean **jj\_2\_8**( int xla )
- *jj\_2\_9*  
private final boolean **jj\_2\_9**( int xla )
- *jj\_3\_1*  
private final boolean **jj\_3\_1**( )
- *jj\_3\_10*  
private final boolean **jj\_3\_10**( )



- *jj-3-11*  
private final boolean **jj\_3\_11**( )
- *jj-3-12*  
private final boolean **jj\_3\_12**( )
- *jj-3-13*  
private final boolean **jj\_3\_13**( )
- *jj-3-14*  
private final boolean **jj\_3\_14**( )
- *jj-3-15*  
private final boolean **jj\_3\_15**( )
- *jj-3-16*  
private final boolean **jj\_3\_16**( )
- *jj-3-17*  
private final boolean **jj\_3\_17**( )
- *jj-3-18*  
private final boolean **jj\_3\_18**( )
- *jj-3-19*  
private final boolean **jj\_3\_19**( )
- *jj-3-2*  
private final boolean **jj\_3\_2**( )
- *jj-3-20*  
private final boolean **jj\_3\_20**( )
- *jj-3-21*  
private final boolean **jj\_3\_21**( )
- *jj-3-22*  
private final boolean **jj\_3\_22**( )
- *jj-3-23*  
private final boolean **jj\_3\_23**( )
- *jj-3-24*  
private final boolean **jj\_3\_24**( )
- *jj-3-25*  
private final boolean **jj\_3\_25**( )
- *jj-3-26*  
private final boolean **jj\_3\_26**( )
- *jj-3-27*  
private final boolean **jj\_3\_27**( )
- *jj-3-28*  
private final boolean **jj\_3\_28**( )
- *jj-3-29*  
private final boolean **jj\_3\_29**( )

- *jj-3-3*  
private final boolean **jj\_3\_3**( )
- *jj-3-30*  
private final boolean **jj\_3\_30**( )
- *jj-3-31*  
private final boolean **jj\_3\_31**( )
- *jj-3-32*  
private final boolean **jj\_3\_32**( )
- *jj-3-33*  
private final boolean **jj\_3\_33**( )
- *jj-3-34*  
private final boolean **jj\_3\_34**( )
- *jj-3-35*  
private final boolean **jj\_3\_35**( )
- *jj-3-36*  
private final boolean **jj\_3\_36**( )
- *jj-3-37*  
private final boolean **jj\_3\_37**( )
- *jj-3-38*  
private final boolean **jj\_3\_38**( )
- *jj-3-39*  
private final boolean **jj\_3\_39**( )
- *jj-3-4*  
private final boolean **jj\_3\_4**( )
- *jj-3-40*  
private final boolean **jj\_3\_40**( )
- *jj-3-41*  
private final boolean **jj\_3\_41**( )
- *jj-3-42*  
private final boolean **jj\_3\_42**( )
- *jj-3-43*  
private final boolean **jj\_3\_43**( )
- *jj-3-44*  
private final boolean **jj\_3\_44**( )
- *jj-3-45*  
private final boolean **jj\_3\_45**( )
- *jj-3-46*  
private final boolean **jj\_3\_46**( )
- *jj-3-47*  
private final boolean **jj\_3\_47**( )

- *jj-3-48*  
private final boolean **jj\_3\_48**( )
- *jj-3-49*  
private final boolean **jj\_3\_49**( )
- *jj-3-5*  
private final boolean **jj\_3\_5**( )
- *jj-3-50*  
private final boolean **jj\_3\_50**( )
- *jj-3-51*  
private final boolean **jj\_3\_51**( )
- *jj-3-52*  
private final boolean **jj\_3\_52**( )
- *jj-3-53*  
private final boolean **jj\_3\_53**( )
- *jj-3-54*  
private final boolean **jj\_3\_54**( )
- *jj-3-55*  
private final boolean **jj\_3\_55**( )
- *jj-3-56*  
private final boolean **jj\_3\_56**( )
- *jj-3-57*  
private final boolean **jj\_3\_57**( )
- *jj-3-58*  
private final boolean **jj\_3\_58**( )
- *jj-3-59*  
private final boolean **jj\_3\_59**( )
- *jj-3-6*  
private final boolean **jj\_3\_6**( )
- *jj-3-7*  
private final boolean **jj\_3\_7**( )
- *jj-3-8*  
private final boolean **jj\_3\_8**( )
- *jj-3-9*  
private final boolean **jj\_3\_9**( )
- *jj-3R-100*  
private final boolean **jj\_3R\_100**( )
- *jj-3R-101*  
private final boolean **jj\_3R\_101**( )
- *jj-3R-102*  
private final boolean **jj\_3R\_102**( )

- *jj\_3R\_103*  
private final boolean **jj\_3R\_103**( )
- *jj\_3R\_104*  
private final boolean **jj\_3R\_104**( )
- *jj\_3R\_105*  
private final boolean **jj\_3R\_105**( )
- *jj\_3R\_106*  
private final boolean **jj\_3R\_106**( )
- *jj\_3R\_107*  
private final boolean **jj\_3R\_107**( )
- *jj\_3R\_108*  
private final boolean **jj\_3R\_108**( )
- *jj\_3R\_109*  
private final boolean **jj\_3R\_109**( )
- *jj\_3R\_110*  
private final boolean **jj\_3R\_110**( )
- *jj\_3R\_111*  
private final boolean **jj\_3R\_111**( )
- *jj\_3R\_112*  
private final boolean **jj\_3R\_112**( )
- *jj\_3R\_113*  
private final boolean **jj\_3R\_113**( )
- *jj\_3R\_114*  
private final boolean **jj\_3R\_114**( )
- *jj\_3R\_115*  
private final boolean **jj\_3R\_115**( )
- *jj\_3R\_116*  
private final boolean **jj\_3R\_116**( )
- *jj\_3R\_117*  
private final boolean **jj\_3R\_117**( )
- *jj\_3R\_118*  
private final boolean **jj\_3R\_118**( )
- *jj\_3R\_119*  
private final boolean **jj\_3R\_119**( )
- *jj\_3R\_120*  
private final boolean **jj\_3R\_120**( )
- *jj\_3R\_121*  
private final boolean **jj\_3R\_121**( )
- *jj\_3R\_122*  
private final boolean **jj\_3R\_122**( )

- *jj\_3R\_123*  
private final boolean **jj\_3R\_123**( )
- *jj\_3R\_124*  
private final boolean **jj\_3R\_124**( )
- *jj\_3R\_125*  
private final boolean **jj\_3R\_125**( )
- *jj\_3R\_126*  
private final boolean **jj\_3R\_126**( )
- *jj\_3R\_127*  
private final boolean **jj\_3R\_127**( )
- *jj\_3R\_128*  
private final boolean **jj\_3R\_128**( )
- *jj\_3R\_129*  
private final boolean **jj\_3R\_129**( )
- *jj\_3R\_130*  
private final boolean **jj\_3R\_130**( )
- *jj\_3R\_131*  
private final boolean **jj\_3R\_131**( )
- *jj\_3R\_132*  
private final boolean **jj\_3R\_132**( )
- *jj\_3R\_133*  
private final boolean **jj\_3R\_133**( )
- *jj\_3R\_134*  
private final boolean **jj\_3R\_134**( )
- *jj\_3R\_135*  
private final boolean **jj\_3R\_135**( )
- *jj\_3R\_136*  
private final boolean **jj\_3R\_136**( )
- *jj\_3R\_137*  
private final boolean **jj\_3R\_137**( )
- *jj\_3R\_138*  
private final boolean **jj\_3R\_138**( )
- *jj\_3R\_139*  
private final boolean **jj\_3R\_139**( )
- *jj\_3R\_140*  
private final boolean **jj\_3R\_140**( )
- *jj\_3R\_141*  
private final boolean **jj\_3R\_141**( )
- *jj\_3R\_142*  
private final boolean **jj\_3R\_142**( )

- *jj-3R\_143*  
private final boolean **jj\_3R\_143**( )
- *jj-3R\_144*  
private final boolean **jj\_3R\_144**( )
- *jj-3R\_145*  
private final boolean **jj\_3R\_145**( )
- *jj-3R\_146*  
private final boolean **jj\_3R\_146**( )
- *jj-3R\_147*  
private final boolean **jj\_3R\_147**( )
- *jj-3R\_148*  
private final boolean **jj\_3R\_148**( )
- *jj-3R\_149*  
private final boolean **jj\_3R\_149**( )
- *jj-3R\_150*  
private final boolean **jj\_3R\_150**( )
- *jj-3R\_151*  
private final boolean **jj\_3R\_151**( )
- *jj-3R\_152*  
private final boolean **jj\_3R\_152**( )
- *jj-3R\_153*  
private final boolean **jj\_3R\_153**( )
- *jj-3R\_154*  
private final boolean **jj\_3R\_154**( )
- *jj-3R\_155*  
private final boolean **jj\_3R\_155**( )
- *jj-3R\_156*  
private final boolean **jj\_3R\_156**( )
- *jj-3R\_157*  
private final boolean **jj\_3R\_157**( )
- *jj-3R\_158*  
private final boolean **jj\_3R\_158**( )
- *jj-3R\_159*  
private final boolean **jj\_3R\_159**( )
- *jj-3R\_160*  
private final boolean **jj\_3R\_160**( )
- *jj-3R\_161*  
private final boolean **jj\_3R\_161**( )
- *jj-3R\_162*  
private final boolean **jj\_3R\_162**( )

- *jj\_3R\_163*  
private final boolean **jj\_3R\_163**( )
- *jj\_3R\_164*  
private final boolean **jj\_3R\_164**( )
- *jj\_3R\_165*  
private final boolean **jj\_3R\_165**( )
- *jj\_3R\_166*  
private final boolean **jj\_3R\_166**( )
- *jj\_3R\_167*  
private final boolean **jj\_3R\_167**( )
- *jj\_3R\_168*  
private final boolean **jj\_3R\_168**( )
- *jj\_3R\_169*  
private final boolean **jj\_3R\_169**( )
- *jj\_3R\_170*  
private final boolean **jj\_3R\_170**( )
- *jj\_3R\_171*  
private final boolean **jj\_3R\_171**( )
- *jj\_3R\_172*  
private final boolean **jj\_3R\_172**( )
- *jj\_3R\_173*  
private final boolean **jj\_3R\_173**( )
- *jj\_3R\_174*  
private final boolean **jj\_3R\_174**( )
- *jj\_3R\_175*  
private final boolean **jj\_3R\_175**( )
- *jj\_3R\_176*  
private final boolean **jj\_3R\_176**( )
- *jj\_3R\_177*  
private final boolean **jj\_3R\_177**( )
- *jj\_3R\_178*  
private final boolean **jj\_3R\_178**( )
- *jj\_3R\_179*  
private final boolean **jj\_3R\_179**( )
- *jj\_3R\_180*  
private final boolean **jj\_3R\_180**( )
- *jj\_3R\_181*  
private final boolean **jj\_3R\_181**( )
- *jj\_3R\_182*  
private final boolean **jj\_3R\_182**( )

- *jj\_3R\_183*  
private final boolean **jj\_3R\_183**( )
- *jj\_3R\_184*  
private final boolean **jj\_3R\_184**( )
- *jj\_3R\_185*  
private final boolean **jj\_3R\_185**( )
- *jj\_3R\_186*  
private final boolean **jj\_3R\_186**( )
- *jj\_3R\_187*  
private final boolean **jj\_3R\_187**( )
- *jj\_3R\_188*  
private final boolean **jj\_3R\_188**( )
- *jj\_3R\_189*  
private final boolean **jj\_3R\_189**( )
- *jj\_3R\_190*  
private final boolean **jj\_3R\_190**( )
- *jj\_3R\_191*  
private final boolean **jj\_3R\_191**( )
- *jj\_3R\_192*  
private final boolean **jj\_3R\_192**( )
- *jj\_3R\_193*  
private final boolean **jj\_3R\_193**( )
- *jj\_3R\_194*  
private final boolean **jj\_3R\_194**( )
- *jj\_3R\_195*  
private final boolean **jj\_3R\_195**( )
- *jj\_3R\_196*  
private final boolean **jj\_3R\_196**( )
- *jj\_3R\_197*  
private final boolean **jj\_3R\_197**( )
- *jj\_3R\_198*  
private final boolean **jj\_3R\_198**( )
- *jj\_3R\_199*  
private final boolean **jj\_3R\_199**( )
- *jj\_3R\_200*  
private final boolean **jj\_3R\_200**( )
- *jj\_3R\_201*  
private final boolean **jj\_3R\_201**( )
- *jj\_3R\_202*  
private final boolean **jj\_3R\_202**( )



- *jj\_3R\_203*  
private final boolean **jj\_3R\_203**( )
- *jj\_3R\_204*  
private final boolean **jj\_3R\_204**( )
- *jj\_3R\_205*  
private final boolean **jj\_3R\_205**( )
- *jj\_3R\_206*  
private final boolean **jj\_3R\_206**( )
- *jj\_3R\_207*  
private final boolean **jj\_3R\_207**( )
- *jj\_3R\_208*  
private final boolean **jj\_3R\_208**( )
- *jj\_3R\_209*  
private final boolean **jj\_3R\_209**( )
- *jj\_3R\_210*  
private final boolean **jj\_3R\_210**( )
- *jj\_3R\_211*  
private final boolean **jj\_3R\_211**( )
- *jj\_3R\_212*  
private final boolean **jj\_3R\_212**( )
- *jj\_3R\_213*  
private final boolean **jj\_3R\_213**( )
- *jj\_3R\_214*  
private final boolean **jj\_3R\_214**( )
- *jj\_3R\_215*  
private final boolean **jj\_3R\_215**( )
- *jj\_3R\_216*  
private final boolean **jj\_3R\_216**( )
- *jj\_3R\_217*  
private final boolean **jj\_3R\_217**( )
- *jj\_3R\_218*  
private final boolean **jj\_3R\_218**( )
- *jj\_3R\_219*  
private final boolean **jj\_3R\_219**( )
- *jj\_3R\_220*  
private final boolean **jj\_3R\_220**( )
- *jj\_3R\_221*  
private final boolean **jj\_3R\_221**( )
- *jj\_3R\_222*  
private final boolean **jj\_3R\_222**( )

- *jj\_3R\_223*  
private final boolean **jj\_3R\_223**( )
- *jj\_3R\_224*  
private final boolean **jj\_3R\_224**( )
- *jj\_3R\_225*  
private final boolean **jj\_3R\_225**( )
- *jj\_3R\_226*  
private final boolean **jj\_3R\_226**( )
- *jj\_3R\_227*  
private final boolean **jj\_3R\_227**( )
- *jj\_3R\_228*  
private final boolean **jj\_3R\_228**( )
- *jj\_3R\_229*  
private final boolean **jj\_3R\_229**( )
- *jj\_3R\_230*  
private final boolean **jj\_3R\_230**( )
- *jj\_3R\_231*  
private final boolean **jj\_3R\_231**( )
- *jj\_3R\_232*  
private final boolean **jj\_3R\_232**( )
- *jj\_3R\_233*  
private final boolean **jj\_3R\_233**( )
- *jj\_3R\_234*  
private final boolean **jj\_3R\_234**( )
- *jj\_3R\_235*  
private final boolean **jj\_3R\_235**( )
- *jj\_3R\_236*  
private final boolean **jj\_3R\_236**( )
- *jj\_3R\_237*  
private final boolean **jj\_3R\_237**( )
- *jj\_3R\_238*  
private final boolean **jj\_3R\_238**( )
- *jj\_3R\_239*  
private final boolean **jj\_3R\_239**( )
- *jj\_3R\_240*  
private final boolean **jj\_3R\_240**( )
- *jj\_3R\_241*  
private final boolean **jj\_3R\_241**( )
- *jj\_3R\_242*  
private final boolean **jj\_3R\_242**( )

- *jj-3R\_243*  
private final boolean **jj\_3R\_243**( )
- *jj-3R\_244*  
private final boolean **jj\_3R\_244**( )
- *jj-3R\_245*  
private final boolean **jj\_3R\_245**( )
- *jj-3R\_246*  
private final boolean **jj\_3R\_246**( )
- *jj-3R\_247*  
private final boolean **jj\_3R\_247**( )
- *jj-3R\_248*  
private final boolean **jj\_3R\_248**( )
- *jj-3R\_249*  
private final boolean **jj\_3R\_249**( )
- *jj-3R\_250*  
private final boolean **jj\_3R\_250**( )
- *jj-3R\_251*  
private final boolean **jj\_3R\_251**( )
- *jj-3R\_252*  
private final boolean **jj\_3R\_252**( )
- *jj-3R\_253*  
private final boolean **jj\_3R\_253**( )
- *jj-3R\_254*  
private final boolean **jj\_3R\_254**( )
- *jj-3R\_255*  
private final boolean **jj\_3R\_255**( )
- *jj-3R\_256*  
private final boolean **jj\_3R\_256**( )
- *jj-3R\_257*  
private final boolean **jj\_3R\_257**( )
- *jj-3R\_258*  
private final boolean **jj\_3R\_258**( )
- *jj-3R\_259*  
private final boolean **jj\_3R\_259**( )
- *jj-3R\_260*  
private final boolean **jj\_3R\_260**( )
- *jj-3R\_261*  
private final boolean **jj\_3R\_261**( )
- *jj-3R\_262*  
private final boolean **jj\_3R\_262**( )

- *jj\_3R\_263*  
private final boolean **jj\_3R\_263**( )
- *jj\_3R\_264*  
private final boolean **jj\_3R\_264**( )
- *jj\_3R\_265*  
private final boolean **jj\_3R\_265**( )
- *jj\_3R\_266*  
private final boolean **jj\_3R\_266**( )
- *jj\_3R\_267*  
private final boolean **jj\_3R\_267**( )
- *jj\_3R\_268*  
private final boolean **jj\_3R\_268**( )
- *jj\_3R\_269*  
private final boolean **jj\_3R\_269**( )
- *jj\_3R\_270*  
private final boolean **jj\_3R\_270**( )
- *jj\_3R\_271*  
private final boolean **jj\_3R\_271**( )
- *jj\_3R\_272*  
private final boolean **jj\_3R\_272**( )
- *jj\_3R\_273*  
private final boolean **jj\_3R\_273**( )
- *jj\_3R\_274*  
private final boolean **jj\_3R\_274**( )
- *jj\_3R\_275*  
private final boolean **jj\_3R\_275**( )
- *jj\_3R\_276*  
private final boolean **jj\_3R\_276**( )
- *jj\_3R\_277*  
private final boolean **jj\_3R\_277**( )
- *jj\_3R\_278*  
private final boolean **jj\_3R\_278**( )
- *jj\_3R\_279*  
private final boolean **jj\_3R\_279**( )
- *jj\_3R\_280*  
private final boolean **jj\_3R\_280**( )
- *jj\_3R\_281*  
private final boolean **jj\_3R\_281**( )
- *jj\_3R\_282*  
private final boolean **jj\_3R\_282**( )

- *jj\_3R\_283*  
private final boolean **jj\_3R\_283**( )
- *jj\_3R\_284*  
private final boolean **jj\_3R\_284**( )
- *jj\_3R\_285*  
private final boolean **jj\_3R\_285**( )
- *jj\_3R\_286*  
private final boolean **jj\_3R\_286**( )
- *jj\_3R\_287*  
private final boolean **jj\_3R\_287**( )
- *jj\_3R\_288*  
private final boolean **jj\_3R\_288**( )
- *jj\_3R\_289*  
private final boolean **jj\_3R\_289**( )
- *jj\_3R\_290*  
private final boolean **jj\_3R\_290**( )
- *jj\_3R\_291*  
private final boolean **jj\_3R\_291**( )
- *jj\_3R\_292*  
private final boolean **jj\_3R\_292**( )
- *jj\_3R\_293*  
private final boolean **jj\_3R\_293**( )
- *jj\_3R\_294*  
private final boolean **jj\_3R\_294**( )
- *jj\_3R\_295*  
private final boolean **jj\_3R\_295**( )
- *jj\_3R\_296*  
private final boolean **jj\_3R\_296**( )
- *jj\_3R\_297*  
private final boolean **jj\_3R\_297**( )
- *jj\_3R\_298*  
private final boolean **jj\_3R\_298**( )
- *jj\_3R\_299*  
private final boolean **jj\_3R\_299**( )
- *jj\_3R\_300*  
private final boolean **jj\_3R\_300**( )
- *jj\_3R\_301*  
private final boolean **jj\_3R\_301**( )
- *jj\_3R\_302*  
private final boolean **jj\_3R\_302**( )

- *jj\_3R\_303*  
private final boolean **jj\_3R\_303**( )
- *jj\_3R\_304*  
private final boolean **jj\_3R\_304**( )
- *jj\_3R\_305*  
private final boolean **jj\_3R\_305**( )
- *jj\_3R\_306*  
private final boolean **jj\_3R\_306**( )
- *jj\_3R\_307*  
private final boolean **jj\_3R\_307**( )
- *jj\_3R\_308*  
private final boolean **jj\_3R\_308**( )
- *jj\_3R\_309*  
private final boolean **jj\_3R\_309**( )
- *jj\_3R\_310*  
private final boolean **jj\_3R\_310**( )
- *jj\_3R\_311*  
private final boolean **jj\_3R\_311**( )
- *jj\_3R\_312*  
private final boolean **jj\_3R\_312**( )
- *jj\_3R\_313*  
private final boolean **jj\_3R\_313**( )
- *jj\_3R\_314*  
private final boolean **jj\_3R\_314**( )
- *jj\_3R\_315*  
private final boolean **jj\_3R\_315**( )
- *jj\_3R\_316*  
private final boolean **jj\_3R\_316**( )
- *jj\_3R\_317*  
private final boolean **jj\_3R\_317**( )
- *jj\_3R\_318*  
private final boolean **jj\_3R\_318**( )
- *jj\_3R\_319*  
private final boolean **jj\_3R\_319**( )
- *jj\_3R\_320*  
private final boolean **jj\_3R\_320**( )
- *jj\_3R\_321*  
private final boolean **jj\_3R\_321**( )
- *jj\_3R\_322*  
private final boolean **jj\_3R\_322**( )

- *jj\_3R\_323*  
private final boolean **jj\_3R\_323**( )
- *jj\_3R\_324*  
private final boolean **jj\_3R\_324**( )
- *jj\_3R\_325*  
private final boolean **jj\_3R\_325**( )
- *jj\_3R\_326*  
private final boolean **jj\_3R\_326**( )
- *jj\_3R\_327*  
private final boolean **jj\_3R\_327**( )
- *jj\_3R\_328*  
private final boolean **jj\_3R\_328**( )
- *jj\_3R\_329*  
private final boolean **jj\_3R\_329**( )
- *jj\_3R\_330*  
private final boolean **jj\_3R\_330**( )
- *jj\_3R\_331*  
private final boolean **jj\_3R\_331**( )
- *jj\_3R\_332*  
private final boolean **jj\_3R\_332**( )
- *jj\_3R\_333*  
private final boolean **jj\_3R\_333**( )
- *jj\_3R\_334*  
private final boolean **jj\_3R\_334**( )
- *jj\_3R\_335*  
private final boolean **jj\_3R\_335**( )
- *jj\_3R\_336*  
private final boolean **jj\_3R\_336**( )
- *jj\_3R\_337*  
private final boolean **jj\_3R\_337**( )
- *jj\_3R\_338*  
private final boolean **jj\_3R\_338**( )
- *jj\_3R\_339*  
private final boolean **jj\_3R\_339**( )
- *jj\_3R\_340*  
private final boolean **jj\_3R\_340**( )
- *jj\_3R\_341*  
private final boolean **jj\_3R\_341**( )
- *jj\_3R\_342*  
private final boolean **jj\_3R\_342**( )

- *jj-3R-343*  
private final boolean **jj\_3R\_343**( )
- *jj-3R-344*  
private final boolean **jj\_3R\_344**( )
- *jj-3R-345*  
private final boolean **jj\_3R\_345**( )
- *jj-3R-346*  
private final boolean **jj\_3R\_346**( )
- *jj-3R-347*  
private final boolean **jj\_3R\_347**( )
- *jj-3R-348*  
private final boolean **jj\_3R\_348**( )
- *jj-3R-349*  
private final boolean **jj\_3R\_349**( )
- *jj-3R-350*  
private final boolean **jj\_3R\_350**( )
- *jj-3R-351*  
private final boolean **jj\_3R\_351**( )
- *jj-3R-352*  
private final boolean **jj\_3R\_352**( )
- *jj-3R-353*  
private final boolean **jj\_3R\_353**( )
- *jj-3R-354*  
private final boolean **jj\_3R\_354**( )
- *jj-3R-355*  
private final boolean **jj\_3R\_355**( )
- *jj-3R-356*  
private final boolean **jj\_3R\_356**( )
- *jj-3R-357*  
private final boolean **jj\_3R\_357**( )
- *jj-3R-358*  
private final boolean **jj\_3R\_358**( )
- *jj-3R-359*  
private final boolean **jj\_3R\_359**( )
- *jj-3R-360*  
private final boolean **jj\_3R\_360**( )
- *jj-3R-361*  
private final boolean **jj\_3R\_361**( )
- *jj-3R-362*  
private final boolean **jj\_3R\_362**( )



- *jj\_3R\_363*  
private final boolean **jj\_3R\_363**( )
- *jj\_3R\_364*  
private final boolean **jj\_3R\_364**( )
- *jj\_3R\_365*  
private final boolean **jj\_3R\_365**( )
- *jj\_3R\_366*  
private final boolean **jj\_3R\_366**( )
- *jj\_3R\_367*  
private final boolean **jj\_3R\_367**( )
- *jj\_3R\_368*  
private final boolean **jj\_3R\_368**( )
- *jj\_3R\_369*  
private final boolean **jj\_3R\_369**( )
- *jj\_3R\_370*  
private final boolean **jj\_3R\_370**( )
- *jj\_3R\_371*  
private final boolean **jj\_3R\_371**( )
- *jj\_3R\_372*  
private final boolean **jj\_3R\_372**( )
- *jj\_3R\_373*  
private final boolean **jj\_3R\_373**( )
- *jj\_3R\_374*  
private final boolean **jj\_3R\_374**( )
- *jj\_3R\_375*  
private final boolean **jj\_3R\_375**( )
- *jj\_3R\_376*  
private final boolean **jj\_3R\_376**( )
- *jj\_3R\_377*  
private final boolean **jj\_3R\_377**( )
- *jj\_3R\_378*  
private final boolean **jj\_3R\_378**( )
- *jj\_3R\_379*  
private final boolean **jj\_3R\_379**( )
- *jj\_3R\_380*  
private final boolean **jj\_3R\_380**( )
- *jj\_3R\_381*  
private final boolean **jj\_3R\_381**( )
- *jj\_3R\_382*  
private final boolean **jj\_3R\_382**( )

- *jj\_3R\_383*  
private final boolean **jj\_3R\_383**( )
- *jj\_3R\_384*  
private final boolean **jj\_3R\_384**( )
- *jj\_3R\_385*  
private final boolean **jj\_3R\_385**( )
- *jj\_3R\_386*  
private final boolean **jj\_3R\_386**( )
- *jj\_3R\_387*  
private final boolean **jj\_3R\_387**( )
- *jj\_3R\_388*  
private final boolean **jj\_3R\_388**( )
- *jj\_3R\_389*  
private final boolean **jj\_3R\_389**( )
- *jj\_3R\_390*  
private final boolean **jj\_3R\_390**( )
- *jj\_3R\_391*  
private final boolean **jj\_3R\_391**( )
- *jj\_3R\_392*  
private final boolean **jj\_3R\_392**( )
- *jj\_3R\_393*  
private final boolean **jj\_3R\_393**( )
- *jj\_3R\_394*  
private final boolean **jj\_3R\_394**( )
- *jj\_3R\_395*  
private final boolean **jj\_3R\_395**( )
- *jj\_3R\_396*  
private final boolean **jj\_3R\_396**( )
- *jj\_3R\_397*  
private final boolean **jj\_3R\_397**( )
- *jj\_3R\_398*  
private final boolean **jj\_3R\_398**( )
- *jj\_3R\_399*  
private final boolean **jj\_3R\_399**( )
- *jj\_3R\_400*  
private final boolean **jj\_3R\_400**( )
- *jj\_3R\_401*  
private final boolean **jj\_3R\_401**( )
- *jj\_3R\_402*  
private final boolean **jj\_3R\_402**( )

- *jj\_3R\_403*  
private final boolean **jj\_3R\_403**( )
- *jj\_3R\_404*  
private final boolean **jj\_3R\_404**( )
- *jj\_3R\_405*  
private final boolean **jj\_3R\_405**( )
- *jj\_3R\_406*  
private final boolean **jj\_3R\_406**( )
- *jj\_3R\_407*  
private final boolean **jj\_3R\_407**( )
- *jj\_3R\_408*  
private final boolean **jj\_3R\_408**( )
- *jj\_3R\_409*  
private final boolean **jj\_3R\_409**( )
- *jj\_3R\_410*  
private final boolean **jj\_3R\_410**( )
- *jj\_3R\_411*  
private final boolean **jj\_3R\_411**( )
- *jj\_3R\_412*  
private final boolean **jj\_3R\_412**( )
- *jj\_3R\_413*  
private final boolean **jj\_3R\_413**( )
- *jj\_3R\_414*  
private final boolean **jj\_3R\_414**( )
- *jj\_3R\_415*  
private final boolean **jj\_3R\_415**( )
- *jj\_3R\_416*  
private final boolean **jj\_3R\_416**( )
- *jj\_3R\_417*  
private final boolean **jj\_3R\_417**( )
- *jj\_3R\_418*  
private final boolean **jj\_3R\_418**( )
- *jj\_3R\_419*  
private final boolean **jj\_3R\_419**( )
- *jj\_3R\_420*  
private final boolean **jj\_3R\_420**( )
- *jj\_3R\_421*  
private final boolean **jj\_3R\_421**( )
- *jj\_3R\_422*  
private final boolean **jj\_3R\_422**( )

- *jj-3R\_423*  
private final boolean **jj\_3R\_423**( )
- *jj-3R\_424*  
private final boolean **jj\_3R\_424**( )
- *jj-3R\_425*  
private final boolean **jj\_3R\_425**( )
- *jj-3R\_426*  
private final boolean **jj\_3R\_426**( )
- *jj-3R\_427*  
private final boolean **jj\_3R\_427**( )
- *jj-3R\_428*  
private final boolean **jj\_3R\_428**( )
- *jj-3R\_429*  
private final boolean **jj\_3R\_429**( )
- *jj-3R\_430*  
private final boolean **jj\_3R\_430**( )
- *jj-3R\_431*  
private final boolean **jj\_3R\_431**( )
- *jj-3R\_432*  
private final boolean **jj\_3R\_432**( )
- *jj-3R\_433*  
private final boolean **jj\_3R\_433**( )
- *jj-3R\_434*  
private final boolean **jj\_3R\_434**( )
- *jj-3R\_435*  
private final boolean **jj\_3R\_435**( )
- *jj-3R\_436*  
private final boolean **jj\_3R\_436**( )
- *jj-3R\_437*  
private final boolean **jj\_3R\_437**( )
- *jj-3R\_438*  
private final boolean **jj\_3R\_438**( )
- *jj-3R\_439*  
private final boolean **jj\_3R\_439**( )
- *jj-3R\_440*  
private final boolean **jj\_3R\_440**( )
- *jj-3R\_441*  
private final boolean **jj\_3R\_441**( )
- *jj-3R\_442*  
private final boolean **jj\_3R\_442**( )

- *jj-3R\_443*  
private final boolean **jj\_3R\_443**( )
- *jj-3R\_444*  
private final boolean **jj\_3R\_444**( )
- *jj-3R\_445*  
private final boolean **jj\_3R\_445**( )
- *jj-3R\_446*  
private final boolean **jj\_3R\_446**( )
- *jj-3R\_447*  
private final boolean **jj\_3R\_447**( )
- *jj-3R\_448*  
private final boolean **jj\_3R\_448**( )
- *jj-3R\_449*  
private final boolean **jj\_3R\_449**( )
- *jj-3R\_450*  
private final boolean **jj\_3R\_450**( )
- *jj-3R\_451*  
private final boolean **jj\_3R\_451**( )
- *jj-3R\_452*  
private final boolean **jj\_3R\_452**( )
- *jj-3R\_453*  
private final boolean **jj\_3R\_453**( )
- *jj-3R\_454*  
private final boolean **jj\_3R\_454**( )
- *jj-3R\_455*  
private final boolean **jj\_3R\_455**( )
- *jj-3R\_456*  
private final boolean **jj\_3R\_456**( )
- *jj-3R\_457*  
private final boolean **jj\_3R\_457**( )
- *jj-3R\_458*  
private final boolean **jj\_3R\_458**( )
- *jj-3R\_459*  
private final boolean **jj\_3R\_459**( )
- *jj-3R\_460*  
private final boolean **jj\_3R\_460**( )
- *jj-3R\_461*  
private final boolean **jj\_3R\_461**( )
- *jj-3R\_462*  
private final boolean **jj\_3R\_462**( )

- *jj\_3R\_463*  
private final boolean **jj\_3R\_463**( )
- *jj\_3R\_464*  
private final boolean **jj\_3R\_464**( )
- *jj\_3R\_465*  
private final boolean **jj\_3R\_465**( )
- *jj\_3R\_466*  
private final boolean **jj\_3R\_466**( )
- *jj\_3R\_467*  
private final boolean **jj\_3R\_467**( )
- *jj\_3R\_468*  
private final boolean **jj\_3R\_468**( )
- *jj\_3R\_469*  
private final boolean **jj\_3R\_469**( )
- *jj\_3R\_470*  
private final boolean **jj\_3R\_470**( )
- *jj\_3R\_471*  
private final boolean **jj\_3R\_471**( )
- *jj\_3R\_472*  
private final boolean **jj\_3R\_472**( )
- *jj\_3R\_473*  
private final boolean **jj\_3R\_473**( )
- *jj\_3R\_474*  
private final boolean **jj\_3R\_474**( )
- *jj\_3R\_475*  
private final boolean **jj\_3R\_475**( )
- *jj\_3R\_476*  
private final boolean **jj\_3R\_476**( )
- *jj\_3R\_477*  
private final boolean **jj\_3R\_477**( )
- *jj\_3R\_478*  
private final boolean **jj\_3R\_478**( )
- *jj\_3R\_479*  
private final boolean **jj\_3R\_479**( )
- *jj\_3R\_480*  
private final boolean **jj\_3R\_480**( )
- *jj\_3R\_481*  
private final boolean **jj\_3R\_481**( )
- *jj\_3R\_482*  
private final boolean **jj\_3R\_482**( )

- *jj\_3R\_483*  
private final boolean **jj\_3R\_483**( )
- *jj\_3R\_484*  
private final boolean **jj\_3R\_484**( )
- *jj\_3R\_485*  
private final boolean **jj\_3R\_485**( )
- *jj\_3R\_486*  
private final boolean **jj\_3R\_486**( )
- *jj\_3R\_487*  
private final boolean **jj\_3R\_487**( )
- *jj\_3R\_488*  
private final boolean **jj\_3R\_488**( )
- *jj\_3R\_489*  
private final boolean **jj\_3R\_489**( )
- *jj\_3R\_490*  
private final boolean **jj\_3R\_490**( )
- *jj\_3R\_491*  
private final boolean **jj\_3R\_491**( )
- *jj\_3R\_492*  
private final boolean **jj\_3R\_492**( )
- *jj\_3R\_493*  
private final boolean **jj\_3R\_493**( )
- *jj\_3R\_494*  
private final boolean **jj\_3R\_494**( )
- *jj\_3R\_495*  
private final boolean **jj\_3R\_495**( )
- *jj\_3R\_496*  
private final boolean **jj\_3R\_496**( )
- *jj\_3R\_497*  
private final boolean **jj\_3R\_497**( )
- *jj\_3R\_498*  
private final boolean **jj\_3R\_498**( )
- *jj\_3R\_499*  
private final boolean **jj\_3R\_499**( )
- *jj\_3R\_500*  
private final boolean **jj\_3R\_500**( )
- *jj\_3R\_501*  
private final boolean **jj\_3R\_501**( )
- *jj\_3R\_502*  
private final boolean **jj\_3R\_502**( )

- *jj\_3R\_503*  
private final boolean **jj\_3R\_503**( )
- *jj\_3R\_504*  
private final boolean **jj\_3R\_504**( )
- *jj\_3R\_505*  
private final boolean **jj\_3R\_505**( )
- *jj\_3R\_506*  
private final boolean **jj\_3R\_506**( )
- *jj\_3R\_507*  
private final boolean **jj\_3R\_507**( )
- *jj\_3R\_508*  
private final boolean **jj\_3R\_508**( )
- *jj\_3R\_509*  
private final boolean **jj\_3R\_509**( )
- *jj\_3R\_510*  
private final boolean **jj\_3R\_510**( )
- *jj\_3R\_511*  
private final boolean **jj\_3R\_511**( )
- *jj\_3R\_512*  
private final boolean **jj\_3R\_512**( )
- *jj\_3R\_513*  
private final boolean **jj\_3R\_513**( )
- *jj\_3R\_514*  
private final boolean **jj\_3R\_514**( )
- *jj\_3R\_515*  
private final boolean **jj\_3R\_515**( )
- *jj\_3R\_516*  
private final boolean **jj\_3R\_516**( )
- *jj\_3R\_517*  
private final boolean **jj\_3R\_517**( )
- *jj\_3R\_518*  
private final boolean **jj\_3R\_518**( )
- *jj\_3R\_519*  
private final boolean **jj\_3R\_519**( )
- *jj\_3R\_520*  
private final boolean **jj\_3R\_520**( )
- *jj\_3R\_521*  
private final boolean **jj\_3R\_521**( )
- *jj\_3R\_522*  
private final boolean **jj\_3R\_522**( )



- *jj\_3R\_523*  
private final boolean **jj\_3R\_523**( )
- *jj\_3R\_524*  
private final boolean **jj\_3R\_524**( )
- *jj\_3R\_525*  
private final boolean **jj\_3R\_525**( )
- *jj\_3R\_526*  
private final boolean **jj\_3R\_526**( )
- *jj\_3R\_527*  
private final boolean **jj\_3R\_527**( )
- *jj\_3R\_528*  
private final boolean **jj\_3R\_528**( )
- *jj\_3R\_529*  
private final boolean **jj\_3R\_529**( )
- *jj\_3R\_530*  
private final boolean **jj\_3R\_530**( )
- *jj\_3R\_531*  
private final boolean **jj\_3R\_531**( )
- *jj\_3R\_532*  
private final boolean **jj\_3R\_532**( )
- *jj\_3R\_533*  
private final boolean **jj\_3R\_533**( )
- *jj\_3R\_534*  
private final boolean **jj\_3R\_534**( )
- *jj\_3R\_535*  
private final boolean **jj\_3R\_535**( )
- *jj\_3R\_536*  
private final boolean **jj\_3R\_536**( )
- *jj\_3R\_537*  
private final boolean **jj\_3R\_537**( )
- *jj\_3R\_538*  
private final boolean **jj\_3R\_538**( )
- *jj\_3R\_539*  
private final boolean **jj\_3R\_539**( )
- *jj\_3R\_540*  
private final boolean **jj\_3R\_540**( )
- *jj\_3R\_541*  
private final boolean **jj\_3R\_541**( )
- *jj\_3R\_542*  
private final boolean **jj\_3R\_542**( )

- *jj\_3R\_543*  
private final boolean **jj\_3R\_543**( )
- *jj\_3R\_544*  
private final boolean **jj\_3R\_544**( )
- *jj\_3R\_545*  
private final boolean **jj\_3R\_545**( )
- *jj\_3R\_546*  
private final boolean **jj\_3R\_546**( )
- *jj\_3R\_547*  
private final boolean **jj\_3R\_547**( )
- *jj\_3R\_548*  
private final boolean **jj\_3R\_548**( )
- *jj\_3R\_549*  
private final boolean **jj\_3R\_549**( )
- *jj\_3R\_550*  
private final boolean **jj\_3R\_550**( )
- *jj\_3R\_551*  
private final boolean **jj\_3R\_551**( )
- *jj\_3R\_552*  
private final boolean **jj\_3R\_552**( )
- *jj\_3R\_553*  
private final boolean **jj\_3R\_553**( )
- *jj\_3R\_554*  
private final boolean **jj\_3R\_554**( )
- *jj\_3R\_555*  
private final boolean **jj\_3R\_555**( )
- *jj\_3R\_556*  
private final boolean **jj\_3R\_556**( )
- *jj\_3R\_557*  
private final boolean **jj\_3R\_557**( )
- *jj\_3R\_558*  
private final boolean **jj\_3R\_558**( )
- *jj\_3R\_559*  
private final boolean **jj\_3R\_559**( )
- *jj\_3R\_560*  
private final boolean **jj\_3R\_560**( )
- *jj\_3R\_561*  
private final boolean **jj\_3R\_561**( )
- *jj\_3R\_562*  
private final boolean **jj\_3R\_562**( )

- *jj\_3R\_563*  
private final boolean **jj\_3R\_563**( )
- *jj\_3R\_564*  
private final boolean **jj\_3R\_564**( )
- *jj\_3R\_565*  
private final boolean **jj\_3R\_565**( )
- *jj\_3R\_566*  
private final boolean **jj\_3R\_566**( )
- *jj\_3R\_567*  
private final boolean **jj\_3R\_567**( )
- *jj\_3R\_568*  
private final boolean **jj\_3R\_568**( )
- *jj\_3R\_569*  
private final boolean **jj\_3R\_569**( )
- *jj\_3R\_570*  
private final boolean **jj\_3R\_570**( )
- *jj\_3R\_571*  
private final boolean **jj\_3R\_571**( )
- *jj\_3R\_572*  
private final boolean **jj\_3R\_572**( )
- *jj\_3R\_573*  
private final boolean **jj\_3R\_573**( )
- *jj\_3R\_574*  
private final boolean **jj\_3R\_574**( )
- *jj\_3R\_575*  
private final boolean **jj\_3R\_575**( )
- *jj\_3R\_576*  
private final boolean **jj\_3R\_576**( )
- *jj\_3R\_577*  
private final boolean **jj\_3R\_577**( )
- *jj\_3R\_578*  
private final boolean **jj\_3R\_578**( )
- *jj\_3R\_579*  
private final boolean **jj\_3R\_579**( )
- *jj\_3R\_580*  
private final boolean **jj\_3R\_580**( )
- *jj\_3R\_581*  
private final boolean **jj\_3R\_581**( )
- *jj\_3R\_582*  
private final boolean **jj\_3R\_582**( )

- *jj\_3R\_583*  
private final boolean **jj\_3R\_583**( )
- *jj\_3R\_584*  
private final boolean **jj\_3R\_584**( )
- *jj\_3R\_585*  
private final boolean **jj\_3R\_585**( )
- *jj\_3R\_586*  
private final boolean **jj\_3R\_586**( )
- *jj\_3R\_587*  
private final boolean **jj\_3R\_587**( )
- *jj\_3R\_588*  
private final boolean **jj\_3R\_588**( )
- *jj\_3R\_589*  
private final boolean **jj\_3R\_589**( )
- *jj\_3R\_590*  
private final boolean **jj\_3R\_590**( )
- *jj\_3R\_591*  
private final boolean **jj\_3R\_591**( )
- *jj\_3R\_592*  
private final boolean **jj\_3R\_592**( )
- *jj\_3R\_593*  
private final boolean **jj\_3R\_593**( )
- *jj\_3R\_594*  
private final boolean **jj\_3R\_594**( )
- *jj\_3R\_595*  
private final boolean **jj\_3R\_595**( )
- *jj\_3R\_596*  
private final boolean **jj\_3R\_596**( )
- *jj\_3R\_597*  
private final boolean **jj\_3R\_597**( )
- *jj\_3R\_598*  
private final boolean **jj\_3R\_598**( )
- *jj\_3R\_599*  
private final boolean **jj\_3R\_599**( )
- *jj\_3R\_600*  
private final boolean **jj\_3R\_600**( )
- *jj\_3R\_601*  
private final boolean **jj\_3R\_601**( )
- *jj\_3R\_602*  
private final boolean **jj\_3R\_602**( )

- *jj\_3R\_603*  
private final boolean **jj\_3R\_603**( )
- *jj\_3R\_604*  
private final boolean **jj\_3R\_604**( )
- *jj\_3R\_605*  
private final boolean **jj\_3R\_605**( )
- *jj\_3R\_606*  
private final boolean **jj\_3R\_606**( )
- *jj\_3R\_607*  
private final boolean **jj\_3R\_607**( )
- *jj\_3R\_608*  
private final boolean **jj\_3R\_608**( )
- *jj\_3R\_609*  
private final boolean **jj\_3R\_609**( )
- *jj\_3R\_610*  
private final boolean **jj\_3R\_610**( )
- *jj\_3R\_611*  
private final boolean **jj\_3R\_611**( )
- *jj\_3R\_612*  
private final boolean **jj\_3R\_612**( )
- *jj\_3R\_613*  
private final boolean **jj\_3R\_613**( )
- *jj\_3R\_614*  
private final boolean **jj\_3R\_614**( )
- *jj\_3R\_615*  
private final boolean **jj\_3R\_615**( )
- *jj\_3R\_616*  
private final boolean **jj\_3R\_616**( )
- *jj\_3R\_617*  
private final boolean **jj\_3R\_617**( )
- *jj\_3R\_618*  
private final boolean **jj\_3R\_618**( )
- *jj\_3R\_619*  
private final boolean **jj\_3R\_619**( )
- *jj\_3R\_620*  
private final boolean **jj\_3R\_620**( )
- *jj\_3R\_621*  
private final boolean **jj\_3R\_621**( )
- *jj\_3R\_622*  
private final boolean **jj\_3R\_622**( )

- *jj\_3R\_623*  
private final boolean **jj\_3R\_623**( )
- *jj\_3R\_624*  
private final boolean **jj\_3R\_624**( )
- *jj\_3R\_625*  
private final boolean **jj\_3R\_625**( )
- *jj\_3R\_626*  
private final boolean **jj\_3R\_626**( )
- *jj\_3R\_627*  
private final boolean **jj\_3R\_627**( )
- *jj\_3R\_628*  
private final boolean **jj\_3R\_628**( )
- *jj\_3R\_629*  
private final boolean **jj\_3R\_629**( )
- *jj\_3R\_630*  
private final boolean **jj\_3R\_630**( )
- *jj\_3R\_631*  
private final boolean **jj\_3R\_631**( )
- *jj\_3R\_632*  
private final boolean **jj\_3R\_632**( )
- *jj\_3R\_633*  
private final boolean **jj\_3R\_633**( )
- *jj\_3R\_634*  
private final boolean **jj\_3R\_634**( )
- *jj\_3R\_635*  
private final boolean **jj\_3R\_635**( )
- *jj\_3R\_636*  
private final boolean **jj\_3R\_636**( )
- *jj\_3R\_637*  
private final boolean **jj\_3R\_637**( )
- *jj\_3R\_638*  
private final boolean **jj\_3R\_638**( )
- *jj\_3R\_639*  
private final boolean **jj\_3R\_639**( )
- *jj\_3R\_640*  
private final boolean **jj\_3R\_640**( )
- *jj\_3R\_641*  
private final boolean **jj\_3R\_641**( )
- *jj\_3R\_642*  
private final boolean **jj\_3R\_642**( )

- *jj\_3R\_643*  
private final boolean **jj\_3R\_643**( )
- *jj\_3R\_74*  
private final boolean **jj\_3R\_74**( )
- *jj\_3R\_75*  
private final boolean **jj\_3R\_75**( )
- *jj\_3R\_76*  
private final boolean **jj\_3R\_76**( )
- *jj\_3R\_77*  
private final boolean **jj\_3R\_77**( )
- *jj\_3R\_78*  
private final boolean **jj\_3R\_78**( )
- *jj\_3R\_79*  
private final boolean **jj\_3R\_79**( )
- *jj\_3R\_80*  
private final boolean **jj\_3R\_80**( )
- *jj\_3R\_81*  
private final boolean **jj\_3R\_81**( )
- *jj\_3R\_82*  
private final boolean **jj\_3R\_82**( )
- *jj\_3R\_83*  
private final boolean **jj\_3R\_83**( )
- *jj\_3R\_84*  
private final boolean **jj\_3R\_84**( )
- *jj\_3R\_85*  
private final boolean **jj\_3R\_85**( )
- *jj\_3R\_86*  
private final boolean **jj\_3R\_86**( )
- *jj\_3R\_87*  
private final boolean **jj\_3R\_87**( )
- *jj\_3R\_88*  
private final boolean **jj\_3R\_88**( )
- *jj\_3R\_89*  
private final boolean **jj\_3R\_89**( )
- *jj\_3R\_90*  
private final boolean **jj\_3R\_90**( )
- *jj\_3R\_91*  
private final boolean **jj\_3R\_91**( )
- *jj\_3R\_92*  
private final boolean **jj\_3R\_92**( )

- *jj\_3R\_93*  
private final boolean **jj\_3R\_93**( )
- *jj\_3R\_94*  
private final boolean **jj\_3R\_94**( )
- *jj\_3R\_95*  
private final boolean **jj\_3R\_95**( )
- *jj\_3R\_96*  
private final boolean **jj\_3R\_96**( )
- *jj\_3R\_97*  
private final boolean **jj\_3R\_97**( )
- *jj\_3R\_98*  
private final boolean **jj\_3R\_98**( )
- *jj\_3R\_99*  
private final boolean **jj\_3R\_99**( )
- *jj\_add\_error\_token*  
private void **jj\_add\_error\_token**( int kind, int pos )
- *jj\_consume\_token*  
private final Token **jj\_consume\_token**( int kind )
- *jj\_ntk*  
private final int **jj\_ntk**( )
- *jj\_rescan\_token*  
private final void **jj\_rescan\_token**( )
- *jj\_save*  
private final void **jj\_save**( int index, int xla )
- *jj\_scan\_token*  
private final boolean **jj\_scan\_token**( int kind )
- *labeledStatement*  
public final Statement **labeledStatement**( )
  - **Usage**
    - \* Parses a labeled statement
  - **See Also**
    - \* `koala.dynamicjava.tree.ContinueTarget` ( in 21.1.1, page 1375)
    - \* `koala.dynamicjava.tree.LabeledStatement` ( in 21.2.64, page 1676)
- *labeledStatementLookahead*  
public final void **labeledStatementLookahead**( )
  - **Usage**
    - \* Used internally for lookahead
- *literal*  
public final Expression **literal**( )



- **Usage**
    - \* Used internally to parse an expression

---
- *literalLookahead*  
`public final void literalLookahead( )`
    - **Usage**
      - \* Used internally for lookahead

---
- *localVariableDeclaration*  
`public final List localVariableDeclaration( )`
    - **Usage**
      - \* Parses one variable declaration
    - **Returns** - a list of node because one variable declaration can contain multiple declarations.
    - **See Also**
      - \* `koala.dynamicjava.tree.VariableDeclaration` ( in 21.2.123, page 1952)

---
- *localVariableDeclarationLookahead*  
`public final void localVariableDeclarationLookahead( )`
    - **Usage**
      - \* Used internally for lookahead

---
- *methodDeclaration*  
`public final MethodDeclaration methodDeclaration( )`
    - **Usage**
      - \* Parses a method declaration
    - **See Also**
      - \* `koala.dynamicjava.tree.MethodDeclaration` ( in 21.2.71, page 1711)

---
- *methodDeclarationLookahead*  
`public final void methodDeclarationLookahead( )`
    - **Usage**
      - \* Used internally for lookahead

---
- *methodDeclarationLookahead2*  
`public final void methodDeclarationLookahead2( )`
    - **Usage**
      - \* Used internally for lookahead

---
- *methodDeclaratorLookahead*  
`public final void methodDeclaratorLookahead( )`
    - **Usage**
      - \* Used internally for lookahead

---

- *modifiers*  
 public final Parser.ModifierFlags **modifiers**( )  
 – **Usage**  
   \* Used internally to parse a modifier  


---
- *modifiersLookahead*  
 public final void **modifiersLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *multiplicativeExpression*  
 public final Expression **multiplicativeExpression**( )  
 – **Usage**  
   \* Used internally to parse an expression  


---
- *multiplicativeExpressionLookahead*  
 public final void **multiplicativeExpressionLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *name*  
 public final List **name**( )  
 – **Usage**  
   \* Parses a name  
 – **Returns** - a list of tree token  
 – **See Also**  
   \* koala.dynamicjava.parser.wrapper.TreeToken ( in 3.2.4, page 136)  


---
- *nameList*  
 public final List **nameList**( )  
 – **Usage**  
   \* Parses a comma separated list of names  
 – **Returns** - a list of list of token  
 – **See Also**  
   \* koala.dynamicjava.parser.wrapper.TreeToken ( in 3.2.4, page 136)  


---
- *nameListLookahead*  
 public final void **nameListLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead  


---
- *nameLookahead*  
 public final void **nameLookahead**( )  
 – **Usage**

---

\* Used internally for lookahead

---

- *packageDeclaration*

**public final PackageDeclaration packageDeclaration( )**

- **Usage**

\* Parses a package declaration

- **See Also**

\* `koala.dynamicjava.tree.PackageDeclaration` ( in 21.2.82, page 1765)

---

- *parseCompilationUnit*

**public final List parseCompilationUnit( )**

- **Usage**

\* Parses a Java compilation unit

- **Returns** - a list of nodes (possibly empty)

- **See Also**

\* `koala.dynamicjava.tree.Node` ( in 21.2.75, page 1732)

---

- *parseStream*

**public final List parseStream( )**

- **Usage**

\* Parses top level statements. This production is not a Java language rule. It is used by DynamicJava.

- **Returns** - a list of nodes (possibly empty)

- **See Also**

\* `koala.dynamicjava.tree.Node` ( in 21.2.75, page 1732)

---

- *postfixExpression*

**public final Expression postfixExpression( )**

- **Usage**

\* Used internally to parse an expression

---

- *postfixExpressionLookahead*

**public final void postfixExpressionLookahead( )**

- **Usage**

\* Used internally for lookahead

---

- *preDecrementExpression*

**public final Expression preDecrementExpression( )**

- **Usage**

\* Used internally to parse an expression

---

- *preDecrementExpressionLookahead*

**public final void preDecrementExpressionLookahead( )**

- **Usage**

\* Used internally for lookahead

- 
- *preIncrementExpression*  
 public final Expression **preIncrementExpression**( )  
 – **Usage**  
 \* Used internally to parse an expression

---

  - *preIncrementExpressionLookahead*  
 public final void **preIncrementExpressionLookahead**( )  
 – **Usage**  
 \* Used internally for lookahead

---

  - *primaryExpression*  
 public final Expression **primaryExpression**( )  
 – **Usage**  
 \* Used internally to parse an expression

---

  - *primaryExpressionLookahead*  
 public final void **primaryExpressionLookahead**( )  
 – **Usage**  
 \* Used internally for lookahead

---

  - *primaryPrefix*  
 public final Expression **primaryPrefix**( )  
 – **Usage**  
 \* Used internally to parse an expression

---

  - *primaryPrefixLookahead*  
 public final void **primaryPrefixLookahead**( )  
 – **Usage**  
 \* Used internally for lookahead

---

  - *primarySuffix*  
 public final Parser.ExpressionSuffix **primarySuffix**( )  
 – **Usage**  
 \* Used internally to parse an expression

---

  - *primarySuffixLookahead*  
 public final void **primarySuffixLookahead**( )  
 – **Usage**  
 \* Used internally for lookahead

---

  - *primitiveType*  
 public final Type **primitiveType**( )  
 – **Usage**

- \* Used internally to parse types
- • *primitiveTypeLookahead*  
 public final void **primitiveTypeLookahead**( )
  - **Usage**
    - \* Used internally for lookahead

---
- *qualifiedName*  
 public final Expression **qualifiedName**( )
  - **Usage**
    - \* Used internally to parse an expression

---
- *ReInit*  
 public void **ReInit**( java.io.InputStream stream )
 

---
- *ReInit*  
 public void **ReInit**( koala.dynamicjava.parser.ParserTokenManager tm )
 

---
- *ReInit*  
 public void **ReInit**( java.io.Reader stream )
 

---
- *relationalExpression*  
 public final Expression **relationalExpression**( )
  - **Usage**
    - \* Used internally to parse an expression

---
- *relationalExpressionLookahead*  
 public final void **relationalExpressionLookahead**( )
  - **Usage**
    - \* Used internally for lookahead

---
- *resultType*  
 public final Type **resultType**( )
  - **Usage**
    - \* Used internally to parse types

---
- *resultTypeLookahead*  
 public final void **resultTypeLookahead**( )
  - **Usage**
    - \* Used internally for lookahead

---
- *returnStatement*  
 public final ReturnStatement **returnStatement**( )
  - **Usage**
    - \* Parses a return statement
  - **See Also**
    - \* koala.dynamicjava.tree.ReturnStatement ( in 21.2.94, page 1820)

- 
- *returnStatementLookahead*  
 public final void **returnStatementLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead

---

  - *setFilename*  
 public void **setFilename**( java.lang.String fname )  
 – **Usage**  
   \* Sets the current filename  
 – **Parameters**  
   \* **fname** - the name of the input file

---

  - *shiftExpression*  
 public final Expression **shiftExpression**( )  
 – **Usage**  
   \* Used internally to parse an expression

---

  - *shiftExpressionLookahead*  
 public final void **shiftExpressionLookahead**( )  
 – **Usage**  
   \* Used internally for lookahead

---

  - *simpleAllocationExpression*  
 public final Allocation **simpleAllocationExpression**( )  
 – **Usage**  
   \* Used internally to parse an expression

---

  - *statement*  
 public final Node **statement**( )  
 – **Usage**  
   \* Parses one statement  
 – **See Also**  
   \* koala.dynamicjava.tree.Node ( in 21.2.75, page 1732)

---

  - *statementExpression*  
 public final Node **statementExpression**( )  
 – **Usage**  
   \* Parses one statement expression  
 – **See Also**  
   \* koala.dynamicjava.tree.Expression ( in 21.2.42, page 1571)

---

  - *statementExpressionList*  
 public final List **statementExpressionList**( )

- **Usage**
    - \* Parses a comma separated list of expression

---
- *statementExpressionListLookahead*  
`public final void statementExpressionListLookahead( )`
    - **Usage**
      - \* Used internally for lookahead

---
  - *statementExpressionLookahead*  
`public final void statementExpressionLookahead( )`
    - **Usage**
      - \* Used internally for lookahead

---
  - *statementLookahead*  
`public final void statementLookahead( )`
    - **Usage**
      - \* Used internally for lookahead

---
  - *superExpression*  
`public final Expression superExpression( )`
    - **Usage**
      - \* Used internally to parse an expression

---
  - *switchStatement*  
`public final SwitchStatement switchStatement( )`
    - **Usage**
      - \* Parses a switch statement
    - **See Also**
      - \* `koala.dynamicjava.tree.SwitchStatement` ( in 21.2.111, page 1900)

---
  - *switchStatementLookahead*  
`public final void switchStatementLookahead( )`
    - **Usage**
      - \* Used internally for lookahead

---
  - *synchronizedStatement*  
`public final SynchronizedStatement synchronizedStatement( )`
    - **Usage**
      - \* Parses a synchronized statement
    - **See Also**
      - \* `koala.dynamicjava.tree.SynchronizedStatement` ( in 21.2.112, page 1905)

---
  - *synchronizedStatementLookahead*  
`public final void synchronizedStatementLookahead( )`

- **Usage**
    - \* Used internally for lookahead

---
- *thisExpression*  
 public final Expression **thisExpression**( )
    - **Usage**
      - \* Used internally to parse an expression

---
  - *throwStatement*  
 public final ThrowStatement **throwStatement**( )
    - **Usage**
      - \* Parses a throw statement
    - **See Also**
      - \* koala.dynamicjava.tree.ThrowStatement ( in 21.2.114, page 1914)

---
  - *throwStatementLookahead*  
 public final void **throwStatementLookahead**( )
    - **Usage**
      - \* Used internally for lookahead

---
  - *topLevelStatement*  
 public final Node **topLevelStatement**( )
    - **Usage**
      - \* Parses one top level statement. This production is not a Java language rule. It is used by DynamicJava.
    - **Returns** - a node
    - **See Also**
      - \* koala.dynamicjava.tree.Node ( in 21.2.75, page 1732)

---
  - *tryStatement*  
 public final TryStatement **tryStatement**( )
    - **Usage**
      - \* Parses a try statement
    - **See Also**
      - \* koala.dynamicjava.tree.TryStatement ( in 21.2.116, page 1920)

---
  - *tryStatementLookahead*  
 public final void **tryStatementLookahead**( )
    - **Usage**
      - \* Used internally for lookahead

---
  - *type*  
 public final Type **type**( )
    - **Usage**
      - \* Used internally to parse types



---

- *typeDeclaration*

public final TypeDeclaration **typeDeclaration**( )

- **Usage**

- \* Parses a type declaration

- **See Also**

- \* koala.dynamicjava.tree.TypeDeclaration ( in 21.2.118, page 1928)

---

- *typeLookahead*

public final void **typeLookahead**( )

- **Usage**

- \* Used internally for lookahead

---

- *unaryExpression*

public final Expression **unaryExpression**( )

- **Usage**

- \* Used internally to parse an expression

---

- *unaryExpressionLookahead*

public final void **unaryExpressionLookahead**( )

- **Usage**

- \* Used internally for lookahead

---

- *unaryExpressionNotPlusMinus*

public final Expression **unaryExpressionNotPlusMinus**( )

- **Usage**

- \* Used internally to parse an expression

---

- *unaryExpressionNotPlusMinusLookahead*

public final void **unaryExpressionNotPlusMinusLookahead**( )

- **Usage**

- \* Used internally for lookahead

---

- *unmodifiedClassDeclaration*

public final ClassDeclaration **unmodifiedClassDeclaration**(  
koala.dynamicjava.parser.Parser.ModifierFlags mf )

- **Usage**

- \* Parses a class declaration without modifier

- **See Also**

- \* koala.dynamicjava.tree.ClassDeclaration ( in 21.2.26, page 1492)

---

- *unmodifiedClassDeclarationLookahead*

public final void **unmodifiedClassDeclarationLookahead**( )

- **Usage**

---

\* Used internally for lookahead

---

• *unmodifiedInterfaceDeclaration*

public final InterfaceDeclaration **unmodifiedInterfaceDeclaration**(  
koala.dynamicjava.parser.Parser.ModifierFlags mf )

– **Usage**

\* Parses a interface declaration without modifier

– **See Also**

\* koala.dynamicjava.tree.InterfaceDeclaration ( in 21.2.62, page 1667)

---

• *unmodifiedInterfaceDeclarationLookahead*

public final void **unmodifiedInterfaceDeclarationLookahead**( )

– **Usage**

\* Used internally for lookahead

---

• *variableDeclaratorIdLookahead*

public final void **variableDeclaratorIdLookahead**( )

– **Usage**

\* Used internally for lookahead

---

• *variableDeclaratorLookahead*

public final void **variableDeclaratorLookahead**( )

– **Usage**

\* Used internally for lookahead

---

• *variableInitializer*

public final Expression **variableInitializer**( )

– **Usage**

\* Parses a variable initializer (ie. an expression or an array initializer)

– **See Also**

\* koala.dynamicjava.tree.Expression ( in 21.2.42, page 1571)

---

• *variableInitializerLookahead*

public final void **variableInitializerLookahead**( )

– **Usage**

\* Used internally for lookahead

---

• *whileStatement*

public final WhileStatement **whileStatement**( )

– **Usage**

\* Parses a while statement

– **See Also**

\* koala.dynamicjava.tree.WhileStatement ( in 21.2.125, page 1962)

---

• *whileStatementLookahead*

public final void **whileStatementLookahead**( )

– **Usage**

\* Used internally for lookahead

## 2.2.4 CLASS **Parser.ArgumentsSuffix**

---

### DECLARATION

---

```
class Parser.ArgumentsSuffix
extends koala.dynamicjava.parser.Parser.ExpressionSuffix
```

### FIELDS

---

- public List arguments
  - The arguments
- public int endLine
  - The end line
- public int endColumn
  - The end column

### CONSTRUCTORS

---

- *Parser.ArgumentsSuffix*  
public **Parser.ArgumentsSuffix**( java.util.List args, int el, int ec )
  - **Usage**
    - \* Creates a new argument suffix
  - **Parameters**
    - \* **args** - the list of argument
    - \* **el** - the end line
    - \* **ec** - the end column

### METHODS

---

- *createExpression*  
public Expression **createExpression**( koala.dynamicjava.tree.Expression prefix, java.util.Iterator it )
  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

METHODS INHERITED FROM CLASS `koala.dynamicjava.parser.Parser.ExpressionSuffix`

---

( in 2.2.12, page 120)

- *createExpression*  

```
public abstract Expression createExpression( koala.dynamicjava.tree.Expression
prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

**2.2.5 CLASS `Parser.ArrayReferenceSuffix`**

---

DECLARATION

---

```
class Parser.ArrayReferenceSuffix
extends koala.dynamicjava.parser.Parser.ExpressionSuffix
```

FIELDS

---

- public Expression expression
  - The expression
- public int endLine
  - The end line
- public int endColumn
  - The end column

CONSTRUCTORS

---

- *Parser.ArrayReferenceSuffix*  

```
public Parser.ArrayReferenceSuffix( koala.dynamicjava.tree.Expression exp,
int el, int ec )
```

  - **Usage**
    - \* Creates a new suffix
  - **Parameters**
    - \* **exp** - the expression
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

- *createExpression*  

```
public Expression createExpression( koala.dynamicjava.tree.Expression
prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

METHODS INHERITED FROM CLASS `koala.dynamicjava.parser.Parser.ExpressionSuffix`

( in 2.2.12, page 120)

- *createExpression*  

```
public abstract Expression createExpression( koala.dynamicjava.tree.Expression
prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

**2.2.6 CLASS `Parser.ClassBody`**

To store a class body

## DECLARATION

```
static class Parser.ClassBody
extends java.lang.Object
```

## CONSTRUCTORS

- *Parser.ClassBody*  

```
Parser.ClassBody( java.util.List l, int el, int ec )
```

**2.2.7 CLASS `Parser.CompileTimeQualifiedName`**

## DECLARATION

```
class Parser.CompileTimeQualifiedName
extends koala.dynamicjava.tree.Expression
implements Parser.CompileTimeExpression
```

## FIELDS

---

- public List identifiers
  - The identifiers (tokens)

## CONSTRUCTORS

---

- *Parser.CompileTimeQualifiedName*  
public **Parser.CompileTimeQualifiedName**( java.util.List ids )
  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **ids** - a list of token

## METHODS

---

- *acceptVisitor*  
public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
  - **Usage**
    - \* Not used
- *createFieldAccess*  
public Expression **createFieldAccess**( )
  - **Usage**
    - \* Creates a runtime qualified name
- *createRuntimeExpression*  
public Expression **createRuntimeExpression**( )
  - **Usage**
    - \* Builds a runtime expression from this compile-time one. This method must be called when the expression has no suffix

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

---

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list. The listener is registered for all properties.

- **Parameters**

- \* `listener` - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

- **Parameters**

- \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName,  
java.lang.Object oldValue, java.lang.Object newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *getBeginColumn*  
`public int getBeginColumn( )`
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  
`public int getBeginLine( )`
  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  
`public int getEndColumn( )`
  - **Usage**
    - \* Returns the end column of this node in the end line

---
- *getEndLine*  
`public int getEndLine( )`
  - **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  
`public String getFilename( )`
  - **Usage**
    - \* Returns the filename. Can be null.

---
- *getProperties*  
`public Set getProperties( )`
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
`public Object getProperty( java.lang.String name )`
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* `name` - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`
  - **Usage**



- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list. This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 2.2.8 CLASS **Parser.CompileTimeSuperExpression**

---

### DECLARATION

---

```
class Parser.CompileTimeSuperExpression
extends koala.dynamicjava.tree.Expression
implements Parser.CompileTimeExpression
```

### FIELDS

---

- public String identifier
  - The identifier from the superclass

### CONSTRUCTORS

---

- *Parser.CompileTimeSuperExpression*  
 public **Parser.CompileTimeSuperExpression**( java.lang.String **ident**, int **bl**, int **bc**, int **el**, int **ec** )
  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **ident** - the identifier from the super class
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

### METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor **visitor** )
    - **Usage**
      - \* Not used
- 
- *createRuntimeExpression*  
 public Expression **createRuntimeExpression**( )
    - **Usage**
      - \* Builds a runtime expression from this compile-time one. This method must be called when the expression has no suffix

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list. The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
 \* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
 \* `oldValue` - The old value of the property.  
 \* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  

```
public int getBeginLine( )
```

  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  

```
public int getEndColumn( )
```

  - **Usage**
    - \* Returns the end column of this node in the end line

---
- *getEndLine*  

```
public int getEndLine( )
```

  - **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  

```
public String getFilename( )
```

  - **Usage**
    - \* Returns the filename. Can be null.

---
- *getProperties*  

```
public Set getProperties( )
```

  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  

```
public Object getProperty( java.lang.String  name )
```

  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`
    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* **name** - the name of the property

---

- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`
    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list. This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`
    - **Usage**
      - \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**
      - \* **propertyName** - The name of the property that was listened on.
      - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
`public void setBeginColumn( int i )`
    - **Usage**
      - \* Sets the begin column

---

- *setBeginLine*  
`public void setBeginLine( int i )`
    - **Usage**
      - \* Sets the begin line

---

- *setEndColumn*  
`public void setEndColumn( int i )`
    - **Usage**
      - \* Sets the end column

---

- *setEndLine*  
`public void setEndLine( int i )`
    - **Usage**
      - \* Sets the end line

---

- *setFilename*  
`public void setFilename( java.lang.String s )`
    - **Usage**
      - \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 2.2.9 CLASS *Parser.DotAllocationSuffix*

---

#### DECLARATION

---

```
class Parser.DotAllocationSuffix
extends koala.dynamicjava.parser.Parser.ExpressionSuffix
```

#### FIELDS

---

- public Allocation expression
  - The allocation expression

#### CONSTRUCTORS

---

- *Parser.DotAllocationSuffix*  
 public **Parser.DotAllocationSuffix**( koala.dynamicjava.tree.Allocation exp )
  - **Usage**
    - \* Creates a new suffix
  - **Parameters**
    - \* **exp** - the expression

#### METHODS

---

- *createExpression*  
 public Expression **createExpression**( koala.dynamicjava.tree.Expression prefix, java.util.Iterator it )
  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

METHODS INHERITED FROM CLASS `koala.dynamicjava.parser.Parser.ExpressionSuffix`

---

( in 2.2.12, page 120)

- *createExpression*

```
public abstract Expression createExpression( koala.dynamicjava.tree.Expression
prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

**2.2.10 CLASS `Parser.DotIdentifierSuffix`**

---

DECLARATION

---

```
class Parser.DotIdentifierSuffix
extends koala.dynamicjava.parser.Parser.ExpressionSuffix
```

FIELDS

---

- public Token identifier
  - The identifier

CONSTRUCTORS

---

- *Parser.DotIdentifierSuffix*

```
public Parser.DotIdentifierSuffix( koala.dynamicjava.parser.Token ident )
```

  - **Usage**
    - \* Creates a new suffix
  - **Parameters**
    - \* **ident** - the identifier

METHODS

---

- *createExpression*

```
public Expression createExpression( koala.dynamicjava.tree.Expression
prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

METHODS INHERITED FROM CLASS `koala.dynamicjava.parser.Parser.ExpressionSuffix`

---

( in 2.2.12, page 120)

- *createExpression*  

```
public abstract Expression createExpression( koala.dynamicjava.tree.Expression
prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

**2.2.11 CLASS `Parser.DotThisSuffix`**

---

DECLARATION

---

```
class Parser.DotThisSuffix
extends koala.dynamicjava.parser.Parser.ExpressionSuffix
```

FIELDS

---

- public int `endLine`
  - The end line
- public int `endColumn`
  - The end column

CONSTRUCTORS

---

- *Parser.DotThisSuffix*  

```
public Parser.DotThisSuffix( int el, int ec )
```

  - **Usage**
    - \* Creates a new suffix
  - **Parameters**
    - \* **el** - the end line
    - \* **ec** - the end column
    - \* **fn** - the file name



## METHODS

- *createExpression*  

```
public Expression createExpression( koala.dynamicjava.tree.Expression
prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

## METHODS INHERITED FROM CLASS koala.dynamicjava.parser.Parser.ExpressionSuffix

( in 2.2.12, page 120)

- *createExpression*  

```
public abstract Expression createExpression( koala.dynamicjava.tree.Expression
prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

## 2.2.12 CLASS Parser.ExpressionSuffix

## DECLARATION

```
abstract class Parser.ExpressionSuffix
extends java.lang.Object
```

## CONSTRUCTORS

- *Parser.ExpressionSuffix*  
**Parser.ExpressionSuffix( )**

## METHODS

- *createExpression*  

```
public abstract Expression createExpression(
koala.dynamicjava.tree.Expression prefix, java.util.Iterator it )
```

  - **Usage**
    - \* Creates an expression node
  - **Parameters**
    - \* **prefix** - the prefix expression
    - \* **it** - an iterator over the suffixes

### 2.2.13 CLASS `Parser.JJCalls`

---

#### DECLARATION

---

```
static final class Parser.JJCalls
extends java.lang.Object
```

#### CONSTRUCTORS

---

- *Parser.JJCalls*  
**Parser.JJCalls**( )

### 2.2.14 CLASS `Parser.ModifierFlags`

---

To store modifier flags

#### DECLARATION

---

```
static class Parser.ModifierFlags
extends java.lang.Object
```

#### CONSTRUCTORS

---

- *Parser.ModifierFlags*  
**Parser.ModifierFlags**( int flags, int bl, int bc )

### 2.2.15 CLASS `ParserTokenManager`

---

#### DECLARATION

---

```
public class ParserTokenManager
extends java.lang.Object
implements ParserConstants
```

#### FIELDS

---

- public static final String jjstrLiteralImages
- 
- public static final String lexStateNames

- 
- `public static final int jjnewLexState`
- 
- `private ASCII_UCodeESC_CharStream input_stream`
- 
- `private final int jjrounds`
- 
- `private final int jjstateSet`
- 

## CONSTRUCTORS

---

- *ParserTokenManager*  
`public ParserTokenManager(  
koala.dynamicjava.parser.ASCII_UCodeESC_CharStream stream )`
- *ParserTokenManager*  
`public ParserTokenManager(  
koala.dynamicjava.parser.ASCII_UCodeESC_CharStream stream, int lexState )`

## METHODS

---

- *getNextToken*  
`public final Token getNextToken( )`
- *jjAddStates*  
`private final void jjAddStates( int start, int end )`
- *jjCanMove\_0*  
`private static final boolean jjCanMove_0( int hiByte, int i1, int i2,  
long l1, long l2 )`
- *jjCanMove\_1*  
`private static final boolean jjCanMove_1( int hiByte, int i1, int i2,  
long l1, long l2 )`
- *jjCheckNAdd*  
`private final void jjCheckNAdd( int state )`
- *jjCheckNAddStates*  
`private final void jjCheckNAddStates( int start )`
- *jjCheckNAddStates*  
`private final void jjCheckNAddStates( int start, int end )`
- *jjCheckNAddTwoStates*  
`private final void jjCheckNAddTwoStates( int state1, int state2 )`

- *jjFillToken*  
private final Token **jjFillToken**( )
- *jjMoveNfa\_0*  
private final int **jjMoveNfa\_0**( int startState, int curPos )
- *jjMoveNfa\_1*  
private final int **jjMoveNfa\_1**( int startState, int curPos )
- *jjMoveStringLiteralDfa0\_0*  
private final int **jjMoveStringLiteralDfa0\_0**( )
- *jjMoveStringLiteralDfa0\_1*  
private final int **jjMoveStringLiteralDfa0\_1**( )
- *jjMoveStringLiteralDfa0\_2*  
private final int **jjMoveStringLiteralDfa0\_2**( )
- *jjMoveStringLiteralDfa0\_3*  
private final int **jjMoveStringLiteralDfa0\_3**( )
- *jjMoveStringLiteralDfa1\_0*  
private final int **jjMoveStringLiteralDfa1\_0**( long active0, long active1 )
- *jjMoveStringLiteralDfa1\_2*  
private final int **jjMoveStringLiteralDfa1\_2**( long active0 )
- *jjMoveStringLiteralDfa1\_3*  
private final int **jjMoveStringLiteralDfa1\_3**( long active0 )
- *jjMoveStringLiteralDfa10\_0*  
private final int **jjMoveStringLiteralDfa10\_0**( long old0, long active0 )
- *jjMoveStringLiteralDfa11\_0*  
private final int **jjMoveStringLiteralDfa11\_0**( long old0, long active0 )
- *jjMoveStringLiteralDfa2\_0*  
private final int **jjMoveStringLiteralDfa2\_0**( long old0, long active0, long old1, long active1 )
- *jjMoveStringLiteralDfa3\_0*  
private final int **jjMoveStringLiteralDfa3\_0**( long old0, long active0, long old1, long active1 )
- *jjMoveStringLiteralDfa4\_0*  
private final int **jjMoveStringLiteralDfa4\_0**( long old0, long active0, long old1, long active1 )
- *jjMoveStringLiteralDfa5\_0*  
private final int **jjMoveStringLiteralDfa5\_0**( long old0, long active0 )
- *jjMoveStringLiteralDfa6\_0*  
private final int **jjMoveStringLiteralDfa6\_0**( long old0, long active0 )
- *jjMoveStringLiteralDfa7\_0*  
private final int **jjMoveStringLiteralDfa7\_0**( long old0, long active0 )
- *jjMoveStringLiteralDfa8\_0*  
private final int **jjMoveStringLiteralDfa8\_0**( long old0, long active0 )

- *jjMoveStringLiteralDfa9\_0*  
`private final int jjMoveStringLiteralDfa9_0( long old0, long active0 )`


---
- *jjStartNfa\_0*  
`private final int jjStartNfa_0( int pos, long active0, long active1 )`


---
- *jjStartNfaWithStates\_0*  
`private final int jjStartNfaWithStates_0( int pos, int kind, int state )`


---
- *jjStopAtPos*  
`private final int jjStopAtPos( int pos, int kind )`


---
- *jjStopStringLiteralDfa\_0*  
`private final int jjStopStringLiteralDfa_0( int pos, long active0, long active1 )`


---
- *MoreLexicalActions*  
`final void MoreLexicalActions( )`


---
- *ReInit*  
`public void ReInit( koala.dynamicjava.parser.ASCII_UCodeESC_CharStream stream )`


---
- *ReInit*  
`public void ReInit( koala.dynamicjava.parser.ASCII_UCodeESC_CharStream stream, int lexState )`


---
- *ReInitRounds*  
`private final void ReInitRounds( )`


---
- *SkipLexicalActions*  
`final void SkipLexicalActions( koala.dynamicjava.parser.Token matchedToken )`


---
- *SwitchTo*  
`public void SwitchTo( int lexState )`

## 2.2.16 CLASS Token

---

Describes the input token stream.

### DECLARATION

---

```
public class Token
extends java.lang.Object
```

### FIELDS

---

- public int kind
  - An integer that describes the kind of this token. This numbering system is determined by JavaCCParser, and a table of these numbers is stored in the file ...Constants.java.

- `public int beginLine`
  - `beginLine` and `beginColumn` describe the position of the first character of this token; `endLine` and `endColumn` describe the position of the last character of this token.
- `public int beginColumn`
  - `beginLine` and `beginColumn` describe the position of the first character of this token; `endLine` and `endColumn` describe the position of the last character of this token.
- `public int endLine`
  - `beginLine` and `beginColumn` describe the position of the first character of this token; `endLine` and `endColumn` describe the position of the last character of this token.
- `public int endColumn`
  - `beginLine` and `beginColumn` describe the position of the first character of this token; `endLine` and `endColumn` describe the position of the last character of this token.
- `public String image`
  - The string image of the token.
- `public Token next`
  - A reference to the next regular (non-special) token from the input stream. If this is the last token from the input stream, or if the token manager has not read tokens beyond this one, this field is set to null. This is true only if this token is also a regular token. Otherwise, see below for a description of the contents of this field.
- `public Token specialToken`
  - This field is used to access special tokens that occur prior to this token, but after the immediately preceding regular (non-special) token. If there are no such special tokens, this field is set to null. When there are more than one such special token, this field refers to the last of these special tokens, which in turn refers to the next previous special token through its `specialToken` field, and so on until the first special token (whose `specialToken` field is null). The next fields of special tokens refer to other special tokens that immediately follow it (without an intervening regular token). If there is no such token, this field is null.

## CONSTRUCTORS

---

- *Token*  
`public Token( )`

## METHODS

---

- *newToken*  
`public static final Token newToken( int ofKind )`
  - **Usage**

- \* Returns a new Token object, by default. However, if you want, you can create and return subclass objects based on the value of ofKind. Simply add the cases to the switch for all those special cases. For example, if you have a subclass of Token called IDToken that you want to create if ofKind is ID, simply add something like :

```
case MyParserConstants.ID : return new IDToken();
```

to the following switch statement. Then you can cast matchedToken variable to the appropriate type and use it in your lexical actions.

- *toString*  

```
public final String toString( )
```

  - **Usage**
    - \* Returns the image.

## 2.2.17 CLASS TokenMgrError

---

### DECLARATION

---

```
public class TokenMgrError
extends java.lang.Error
```

### CONSTRUCTORS

---

- *TokenMgrError*  

```
public TokenMgrError( )
```
- *TokenMgrError*  

```
public TokenMgrError( boolean EOFSeen, int lexState, int errorLine,
int errorColumn, java.lang.String errorAfter, char curChar, int reason
)
```
- *TokenMgrError*  

```
public TokenMgrError( java.lang.String message, int reason )
```

### METHODS

---

- *addEscapes*  

```
protected static final String addEscapes( java.lang.String str )
```

  - **Usage**
    - \* Replaces unprintable characters by their escaped (or unicode escaped) equivalents in the given string
- *getMessage*  

```
public String getMessage( )
```

### – Usage

- \* You can also modify the body of this method to customize your error messages. For example, cases like LOOP\_DETECTED and INVALID\_LEXICAL\_STATE are not of end-users concern, so you can return something like :

”Internal Error : Please file a bug report .... ”

from this method for such cases in the release version of your parser.

---

### • *LexicalError*

```
private static final String LexicalError( boolean EOFSeen, int lexState,
int errorLine, int errorColumn, java.lang.String errorAfter, char
curChar )
```

### – Usage

- \* Returns a detailed message for the Error when it is thrown by the token manager to indicate a lexical error. Parameters : EOFSeen : indicates if EOF caused the lexicl error curLexState : lexical state in which this error occured errorLine : line number when the error occured errorColumn : column number when the error occured errorAfter : prefix that was seen before this error occured curchar : the offending character Note: You can customize the lexical error message by modifying this method.

## METHODS INHERITED FROM CLASS java.lang.Error

---

## METHODS INHERITED FROM CLASS java.lang.Throwable

---

- *fillInStackTrace*  
public synchronized native Throwable fillInStackTrace( )
- *getCause*  
public Throwable getCause()
- *getLocalizedMessage*  
public String getLocalizedMessage( )
- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )



- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

## Chapter 3

### Package

### koala.dynamicjava.parser.wrapper

*Package Contents*

*Page*

---

#### Interfaces

<b>ParserFactory</b> .....	130
<i>The classes that implements this interface represent parser factories.</i>	
<b>SourceCodeParser</b> .....	130
<i>The classes that implements this interface represent parsers.</i>	

#### Classes

<b>JavaCCParser</b> .....	131
<i>The instances of this class represents a parser generated with JavaCC.</i>	
<b>JavaCCParserFactory</b> .....	133
<i>This class implements a JavaCCParser factory</i>	
<b>ParseError</b> .....	134
<i>This error is thrown when an unexpected error append while parsing a statement</i>	
<b>TreeToken</b> .....	136
<i>This class represents the token managed by the syntax tree</i>	

---

## 3.1 Interfaces

### 3.1.1 INTERFACE ParserFactory

---

The classes that implements this interface represent parser factories.

#### DECLARATION

---

public interface ParserFactory
--------------------------------

#### METHODS

---

- *createParser*

```
public SourceCodeParser createParser( java.io.InputStream is,  
java.lang.String fname )
```

- **Usage**

- \* Creates a new parser and returns it

- **Parameters**

- \* **is** - the input stream
    - \* **fname** - the file name

---

- *createParser*

```
public SourceCodeParser createParser( java.io.Reader r, java.lang.String  
fname )
```

- **Usage**

- \* Creates a new parser and returns it

- **Parameters**

- \* **r** - the reader
    - \* **fname** - the file name

### 3.1.2 INTERFACE SourceCodeParser

---

The classes that implements this interface represent parsers.

#### DECLARATION

---

public interface SourceCodeParser
-----------------------------------

METHODS

---

• *createParser*

```
public SourceCodeParser createParser( java.io.InputStream is,
    java.lang.String fname )
```

– **Usage**

\* Creates a new parser and returns it

– **Parameters**

\* **is** - the input stream  
 \* **fname** - the file name

---

• *createParser*

```
public SourceCodeParser createParser( java.io.Reader r, java.lang.String
    fname )
```

– **Usage**

\* Creates a new parser and returns it

– **Parameters**

\* **r** - the reader  
 \* **fname** - the file name

---

• *parseCompilationUnit*

```
public List parseCompilationUnit( )
```

– **Usage**

\* Parses a library file

– **See Also**

\* `koala.dynamicjava.tree.Node` ( in 21.2.75, page 1732)

---

• *parseStream*

```
public List parseStream( )
```

– **Usage**

\* Parses top level statements

– **Returns** - a list of nodes– **See Also**

\* `koala.dynamicjava.tree.Node` ( in 21.2.75, page 1732)

## 3.2 Classes

### 3.2.1 CLASS JavaCCParser

---

The instances of this class represents a parser generated with JavaCC.

## DECLARATION

---

```
public class JavaCCParser
extends java.lang.Object
implements SourceCodeParser
```

## FIELDS

---

- private Parser parser
  - The parser

## CONSTRUCTORS

---

- *JavaCCParser*  
public **JavaCCParser**( java.io.InputStream is, java.lang.String fname )
    - **Usage**
      - \* Creates a new JavaCCParser
    - **Parameters**
      - \* **is** - the input stream
      - \* **fname** - the file name
- 
- *JavaCCParser*  
public **JavaCCParser**( java.io.Reader r, java.lang.String fname )
    - **Usage**
      - \* Creates a new JavaCCParser
    - **Parameters**
      - \* **r** - the reader
      - \* **fname** - the file name

## METHODS

---

- *createParser*  
public SourceCodeParser **createParser**( java.io.InputStream is, java.lang.String fname )
    - **Usage**
      - \* Creates a new parser and returns it
    - **Parameters**
      - \* **is** - the input stream
      - \* **fname** - the file name
- 
- *createParser*  
public SourceCodeParser **createParser**( java.io.Reader r, java.lang.String fname )

- **Usage**
  - \* Creates a new parser and returns it
- **Parameters**
  - \* **r** - the reader
  - \* **fname** - the file name

---

- *parseCompilationUnit*

```
public List parseCompilationUnit( )
```

- **Usage**
  - \* Parses a library file
- **See Also**
  - \* `koala.dynamicjava.tree.Node` ( in 21.2.75, page 1732)

---

- *parseStream*

```
public List parseStream( )
```

- **Usage**
  - \* Parses top level statements
- **Returns** - a list of nodes
- **See Also**
  - \* `koala.dynamicjava.tree.Node` ( in 21.2.75, page 1732)

### 3.2.2 CLASS JavaCCParserFactory

---

This class implements a JavaCCParser factory

#### DECLARATION

---

```
public class JavaCCParserFactory
extends java.lang.Object
implements ParserFactory
```

#### CONSTRUCTORS

---

- *JavaCCParserFactory*

```
public JavaCCParserFactory( )
```

#### METHODS

---

- *createParser*

```
public SourceCodeParser createParser( java.io.InputStream is,
java.lang.String fname )
```

  - **Usage**
    - \* Creates a new parser and returns it

---

– **Parameters**

- \* **is** - the input stream
  - \* **fname** - the file name
- 

- *createParser*

```
public SourceCodeParser createParser( java.io.Reader  r, java.lang.String
fname )
```

– **Usage**

- \* Creates a new parser and returns it

– **Parameters**

- \* **r** - the reader
- \* **fname** - the file name

### 3.2.3 CLASS **ParseError**

---

This error is thrown when an unexpected error append while parsing a statement

#### DECLARATION

---

```
public class ParseError
extends java.lang.Error
```

#### SERIALIZABLE FIELDS

---

- private String filename
  - The file name
- private int line
  - The line in the source code where the error occurred
- private int column
  - The column in the source code where the error occurred

#### FIELDS

---

- private String filename
  - The file name
- private int line
  - The line in the source code where the error occurred
- private int column
  - The column in the source code where the error occurred

CONSTRUCTORS

---

• *ParseError*

```
public ParseError( )
```

– **Usage**

\* Constructs an **ExecutionError** with no detail message.

---

• *ParseError*

```
public ParseError( java.lang.String s )
```

– **Usage**

\* Constructs an **ExecutionError** with the specified detail message.

– **Parameters**

\* **s** - the detail message.

---

• *ParseError*

```
public ParseError( java.lang.String s, java.lang.String fn, int l, int c  
 )
```

– **Usage**

\* Constructs an **ExecutionError** with the specified detail message, filename, line and column.

– **Parameters**

\* **s** - the detail message.  
\* **fn** - the file name.  
\* **l** - the line in the source code.  
\* **c** - the column in the source code.

METHODS

---

• *getColumn*

```
public int getColumn( )
```

– **Usage**

\* Returns the column in the source code where the error occurred

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

\* Returns the name of the source file

---

• *getLine*

```
public int getLine( )
```

– **Usage**

\* Returns the line in the source code where the error occurred



METHODS INHERITED FROM CLASS `java.lang.Error`METHODS INHERITED FROM CLASS `java.lang.Throwable`

- *fillInStackTrace*  
public synchronized native Throwable fillInStackTrace( )
- *getCause*  
public Throwable getCause( )
- *getLocalizedMessage*  
public String getLocalizedMessage( )
- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

3.2.4 CLASS *TreeToken*

This class represents the token managed by the syntax tree

DECLARATION

---

```
public class TreeToken
extends java.lang.Object
implements koala.dynamicjava.tree.IdentifierToken
```

FIELDS

---

- private Token token
  - The implementation

CONSTRUCTORS

---

- *TreeToken*  
**public TreeToken( koala.dynamicjava.parser.Token t )**
  - **Usage**
    - \* Creates a new tree token
  - **Parameters**
    - \* **t** - the parser token

METHODS

---

- *beginColumn*  
**public int beginColumn( )**
  - **Usage**
    - \* Returns the column number where the beginning of the token was found in the source file

---
- *beginLine*  
**public int beginLine( )**
  - **Usage**
    - \* Returns the line number where the beginning of the token was found in the source file

---
- *endColumn*  
**public int endColumn( )**
  - **Usage**
    - \* Returns the column number where the end of the token was found in the source file

---
- *endLine*  
**public int endLine( )**
  - **Usage**

\* Returns the line number where the end of the token was found in the source file

---

- *getToken*

**public Token getToken( )**

– **Usage**

\* Returns the underlying token

---

- *image*

**public String image( )**

– **Usage**

\* Returns the representation of the identifier

## Chapter 4

# Package jeliot.launcher

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Classes</b>	
<b>Launcher</b> ..... 140	
<i>Launcher creates a new thread to handle the DynamicJava Interpreter's interpretation.</i>	
<hr/>	

## 4.1 Classes

### 4.1.1 CLASS Launcher

---

Launcher creates a new thread to handle the DynamicJava Interpreter's interpretation. It also handles the piped streams between the two threads one running the Jeliot's GUI and other one DynamicJava.

#### DECLARATION

---

```
public class Launcher
extends java.lang.Thread
```

#### FIELDS

---

- private PipedWriter pipedWriter
  -
- private PipedReader pipedReader
  -
- private PrintWriter writer
  -
- private BufferedReader reader
  -
- private boolean running
  -
- private PipedWriter putInput
  - Pipe communicating Director ->DynamicJava For Input Requests!!!!!!
- private PipedReader getInput
  -
- private PrintWriter inputWriter
  -
- private BufferedReader inputReader
  -
- private String methodCall
  -
- private Reader r
  -

- private Interpreter interpreter

—

- private boolean compiling

—

## CONSTRUCTORS

---

- *Launcher*  
 public **Launcher**( java.io.Reader input )  
 — **Parameters**  
 \* input -

## METHODS

---

- *compile*  
 public void **compile**( )  
 —
- *createInterpreter*  
 protected Interpreter **createInterpreter**( )  
 — **Returns** -  
 —
- *getInputWriter*  
 public PrintWriter **getInputWriter**( )  
 — **Returns** -  
 —
- *getReader*  
 public BufferedReader **getReader**( )  
 — **Returns** -  
 —
- *getWriter*  
 public PrintWriter **getWriter**( )  
 — **Returns** -  
 —
- *makePipedStreams*  
 public void **makePipedStreams**( )  
 —
- *run*  
 public void **run**( )  
 —
- *setCompiling*  
 public void **setCompiling**( boolean value )  
 — **Parameters**  
 \* value -  
 —

- *setMethodCall*  
 public void **setMethodCall**( java.lang.String methodCall )  
 – **Parameters**  
 \* methodCall -  


---
- *stopThread*  
 public void **stopThread**( )

#### METHODS INHERITED FROM CLASS java.lang.Thread

---

- *<clinit>*  
 static void **<clinit>**( )  


---
- *activeCount*  
 public static int **activeCount**( )  


---
- *blockedOn*  
 private void **blockedOn**( sun.nio.ch.Interruptible )  


---
- *checkAccess*  
 public final void **checkAccess**( )  


---
- *countStackFrames*  
 public native int **countStackFrames**( )  


---
- *currentThread*  
 public static native Thread **currentThread**( )  


---
- *destroy*  
 public void **destroy**( )  


---
- *dumpStack*  
 public static void **dumpStack**( )  


---
- *enumerate*  
 public static int **enumerate**( java.lang.Thread [] )  


---
- *exit*  
 private void **exit**( )  


---
- *getContextClassLoader*  
 public ClassLoader **getContextClassLoader**( )  


---
- *getName*  
 public final String **getName**( )  


---
- *getPriority*  
 public final int **getPriority**( )  


---
- *getThreadGroup*  
 public final ThreadGroup **getThreadGroup**( )  


---
- *holdsLock*  
 public static native boolean **holdsLock**( java.lang.Object )  


---
- *init*  
 private void **init**( java.lang.ThreadGroup , java.lang.Runnable , java.lang.String , long )  


---
- *interrupt*  
 public void **interrupt**( )  


---
- *interrupt0*  
 private native void **interrupt0**( )  


---
- *interrupted*  
 public static boolean **interrupted**( )  


---

- *isAlive*  
public final native boolean isAlive( )
- *isDaemon*  
public final boolean isDaemon( )
- *isInterrupted*  
public boolean isInterrupted( )
- *isInterrupted*  
private native boolean isInterrupted( boolean )
- *join*  
public final void join( )
- *join*  
public final synchronized void join( long )
- *join*  
public final synchronized void join( long , int )
- *nextThreadNum*  
private static synchronized int nextThreadNum( )
- *registerNatives*  
private static native void registerNatives( )
- *resume*  
public final void resume( )
- *resume0*  
private native void resume0( )
- *run*  
public void run( )
- *setContextClassLoader*  
public void setContextClassLoader( java.lang.ClassLoader )
- *setDaemon*  
public final void setDaemon( boolean )
- *setName*  
public final void setName( java.lang.String )
- *setPriority*  
public final void setPriority( int )
- *setPriority0*  
private native void setPriority0( int )
- *sleep*  
public static native void sleep( long )
- *sleep*  
public static void sleep( long , int )
- *start*  
public synchronized native void start( )
- *stop*  
public final void stop( )
- *stop*  
public final synchronized void stop( java.lang.Throwable )
- *stop0*  
private native void stop0( java.lang.Object )
- *suspend*  
public final void suspend( )
- *suspend0*  
private native void suspend0( )
- *toString*  
public String toString( )
- *yield*  
public static native void yield( )



## Chapter 5

# Package `koala.dynamicjava.tree.visitor`

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Interfaces</b>	
<b>Visitor</b> .....	145
<i>This interface contains the methods a visitor of the AST must implement</i>	
<b>Classes</b>	
<b>VisitorObject</b> .....	160
<i>This class implements all the methods of Visitor but do nothing (it returns null at each call to 'visit').</i>	
<hr/>	

## 5.1 Interfaces

### 5.1.1 INTERFACE Visitor

This interface contains the methods a visitor of the AST must implement

#### DECLARATION

```
public interface Visitor
```

#### METHODS

- *visit*  

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

  - **Usage**  
 \* Visits a AddAssignExpression
  - **Parameters**  
 \* `node` - the node to visit
- *visit*  

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

  - **Usage**  
 \* Visits a AddExpression
  - **Parameters**  
 \* `node` - the node to visit
- *visit*  

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

  - **Usage**  
 \* Visits a AndExpression
  - **Parameters**  
 \* `node` - the node to visit
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ArrayAccess node )
```

  - **Usage**  
 \* Visits a ArrayAccess
  - **Parameters**  
 \* `node` - the node to visit
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

  - **Usage**

- \* Visits an ArrayAllocation
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.ArrayInitializer node )
  - **Usage**
  - \* Visits an ArrayInitializer
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.ArrayType node )
  - **Usage**
  - \* Visits a ArrayType
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )
  - **Usage**
  - \* Visits a BitAndAssignExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.BitAndExpression node )
  - **Usage**
  - \* Visits a BitAndExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )
  - **Usage**
  - \* Visits a BitOrAssignExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.BitOrExpression node )
  - **Usage**
  - \* Visits a BitOrExpression
  - **Parameters**
  - \* `node` - the node to visit

---

\* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.BlockStatement node )

- Usage

- \* Visits a BlockStatement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.BreakStatement node )

- Usage

- \* Visits a BreakStatement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.CastExpression node )

- Usage

- \* Visits a CastExpression

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.CatchStatement node )

- Usage

- \* Visits a CatchStatement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ClassAllocation node )

- Usage

- \* Visits a ClassAllocation

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ClassDeclaration node )

- Usage

- \* Visits a ClassDeclaration

- Parameters

- \* node - the node to visit

---

- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ClassInitializer node )  
 – **Usage**  
   \* Visits a ClassInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ComplementExpression node )  
 – **Usage**  
   \* Visits a ComplementExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ConditionalExpression node )  
 – **Usage**  
   \* Visits a ConditionalExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ConstructorDeclaration node )  
 – **Usage**  
   \* Visits a ConstructorDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ConstructorInvocation node )  
 – **Usage**  
   \* Visits a ConstructorInvocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ContinueStatement node )  
 – **Usage**  
   \* Visits a ContinueStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.DivideAssignExpression node )

- **Usage**
    - \* Visits a DivideAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideExpression node )
```

  - **Usage**
    - \* Visits a DivideExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DoStatement node )
```

  - **Usage**
    - \* Visits a DoStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.EmptyStatement node )
```

  - **Usage**
    - \* Visits an EmptyStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.EqualExpression node )
```

  - **Usage**
    - \* Visits a EqualExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
```

  - **Usage**
    - \* Visits a ExclusiveOrAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
```

  - **Usage**

- \* Visits a ExclusiveOrExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.FieldDeclaration node )`
  - **Usage**
  - \* Visits a FieldDeclaration
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.FormalParameter node )`
  - **Usage**
  - \* Visits a FormalParameter
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.ForStatement node )`
  - **Usage**
  - \* Visits a ForStatement
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.FunctionCall node )`
  - **Usage**
  - \* Visits a FunctionCall
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.GreaterExpression node )`
  - **Usage**
  - \* Visits a GreaterExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )`
  - **Usage**
  - \* Visits a GreaterOrEqualExpression
  - **Parameters**

- 
- \* node - the node to visit

---

    - *visit*

public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )

      - Usage
        - \* Visits a IfThenElseStatement
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.IfThenStatement node )

      - Usage
        - \* Visits a IfThenStatement
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.ImportDeclaration node )

      - Usage
        - \* Visits an ImportDeclaration
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.InnerAllocation node )

      - Usage
        - \* Visits an InnerAllocation
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )

      - Usage
        - \* Visits an InnerClassAllocation
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.InstanceInitializer node )

      - Usage
        - \* Visits a InstanceInitializer
      - Parameters
        - \* node - the node to visit

---



- *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )  
 – **Usage**  
   \* Visits an InstanceOfExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )  
 – **Usage**  
   \* Visits a InterfaceDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.LabeledStatement node )  
 – **Usage**  
   \* Visits a LabeledStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )  
 – **Usage**  
   \* Visits a LessExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )  
 – **Usage**  
   \* Visits a LessOrEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.Literal node )  
 – **Usage**  
   \* Visits a Literal  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.MethodDeclaration node )

- **Usage**
    - \* Visits a MethodDeclaration
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MinusExpression node )
```

    - **Usage**
      - \* Visits a MinusExpression
    - **Parameters**
      - \* **node** - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

      - **Usage**
        - \* Visits a MultiplyAssignExpression
      - **Parameters**
        - \* **node** - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

        - **Usage**
          - \* Visits a MultiplyExpression
        - **Parameters**
          - \* **node** - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

          - **Usage**
            - \* Visits a NotEqualExpression
          - **Parameters**
            - \* **node** - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.NotExpression node )
```

            - **Usage**
              - \* Visits a NotExpression
            - **Parameters**
              - \* **node** - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
```

              - **Usage**
                - \* Visits an ObjectFieldAccess

- **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
```

    - **Usage**
      - \* Visits an ObjectMethodCall
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.OrExpression node )
```

    - **Usage**
      - \* Visits a OrExpression
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
```

    - **Usage**
      - \* Visits an PackageDeclaration
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.PlusExpression node )
```

    - **Usage**
      - \* Visits a PlusExpression
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement node )
```

    - **Usage**
      - \* Visits a PostDecrement
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement node )
```

    - **Usage**
      - \* Visits a PostIncrement
    - **Parameters**
      - \* `node` - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.PreDecrement  node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.PreIncrement  node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.PrimitiveType  node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.QualifiedName  node )
```

  - **Usage**
    - \* Visits a QualifiedName
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ReferenceType  node )
```

  - **Usage**
    - \* Visits a ReferenceType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression  node )
```

  - **Usage**
    - \* Visits a RemainderAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.RemainderExpression  node )
```

- **Usage**
    - \* Visits a RemainderExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReturnStatement node )
```

  - **Usage**
    - \* Visits a ReturnStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
```

  - **Usage**
    - \* Visits a ShiftLeftAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
```

  - **Usage**
    - \* Visits a ShiftLeftExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
```

  - **Usage**
    - \* Visits a ShiftRightAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
```

  - **Usage**
    - \* Visits a ShiftRightExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
```

  - **Usage**

- \* Visits a SimpleAllocation
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )
  - **Usage**
  - \* Visits a SimpleAssignExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )
  - **Usage**
  - \* Visits a StaticFieldAccess
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.StaticMethodCall node )
  - **Usage**
  - \* Visits a StaticMethodCall
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
  - **Usage**
  - \* Visits a SubtractAssignExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.SubtractExpression node )
  - **Usage**
  - \* Visits a SubtractExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )
  - **Usage**
  - \* Visits a SuperFieldAccess
  - **Parameters**

- 
- \* node - the node to visit

---

    - *visit*

public Object visit( koala.dynamicjava.tree.SuperMethodCall node )

      - Usage
      - \* Visits a SuperMethodCall
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.SwitchBlock node )

      - Usage
      - \* Visits a SwitchBlock
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.SwitchStatement node )

      - Usage
      - \* Visits a SwitchStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )

      - Usage
      - \* Visits a SynchronizedStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ThisExpression node )

      - Usage
      - \* Visits a ThisExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ThrowStatement node )

      - Usage
      - \* Visits a ThrowStatement
      - Parameters
      - \* node - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.TryStatement  node )
```

  - **Usage**  
 \* Visits a TryStatement
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.TypeExpression  node )
```

  - **Usage**  
 \* Visits a TypeExpression
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit(
koala.dynamicjava.tree.UnsignedShiftRightAssignExpression  node )
```

  - **Usage**  
 \* Visits a UnsignedShiftRightAssignExpression
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression
node )
```

  - **Usage**  
 \* Visits a UnsignedShiftRightExpression
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration  node )
```

  - **Usage**  
 \* Visits a VariableDeclaration
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.WhileStatement  node )
```

  - **Usage**  
 \* Visits a WhileStatement
  - **Parameters**  
 \* `node` - the node to visit



## 5.2 Classes

### 5.2.1 CLASS VisitorObject

This class implements all the methods of Visitor but do nothing (it returns null at each call to 'visit'). This class exists as convenience for creating visitor objects

#### DECLARATION

```
public class VisitorObject
extends java.lang.Object
implements Visitor
```

#### CONSTRUCTORS

- *VisitorObject*  
**public VisitorObject( )**

#### METHODS

- *visit*  
**public Object visit( koala.dynamicjava.tree.AddAssignExpression node )**  
 – **Usage**  
   \* Visits an AddAssignExpression  
 – **Parameters**  
   \* **node** - the node to visit
- *visit*  
**public Object visit( koala.dynamicjava.tree.AddExpression node )**  
 – **Usage**  
   \* Visits a AddExpression  
 – **Parameters**  
   \* **node** - the node to visit
- *visit*  
**public Object visit( koala.dynamicjava.tree.AndExpression node )**  
 – **Usage**  
   \* Visits a AndExpression  
 – **Parameters**  
   \* **node** - the node to visit
- *visit*  
**public Object visit( koala.dynamicjava.tree.ArrayAccess node )**

- **Usage**
    - \* Visits a `ArrayAccess`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

  - **Usage**
    - \* Visits an `ArrayAllocation`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayInitializer node )
```

  - **Usage**
    - \* Visits an `ArrayInitializer`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayType node )
```

  - **Usage**
    - \* Visits a `ArrayType`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )
```

  - **Usage**
    - \* Visits a `BitAndAssignExpression`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndExpression node )
```

  - **Usage**
    - \* Visits a `BitAndExpression`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )
```

  - **Usage**
    - \* Visits a `BitOrAssignExpression`

- **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrExpression  node )
```

    - **Usage**
      - \* Visits a BitOrExpression
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.BlockStatement  node )
```

      - **Usage**
        - \* Visits a BlockStatement
      - **Parameters**
        - \* `node` - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.BreakStatement  node )
```

        - **Usage**
          - \* Visits a BreakStatement
        - **Parameters**
          - \* `node` - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.CastExpression  node )
```

          - **Usage**
            - \* Visits a CastExpression
          - **Parameters**
            - \* `node` - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.CatchStatement  node )
```

            - **Usage**
              - \* Visits a CatchStatement
            - **Parameters**
              - \* `node` - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.ClassAllocation  node )
```

              - **Usage**
                - \* Visits an ClassAllocation
              - **Parameters**
                - \* `node` - the node to visit

---

- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ClassDeclaration node )  
 – **Usage**  
   \* Visits a ClassDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ClassInitializer node )  
 – **Usage**  
   \* Visits a ClassInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ComplementExpression node )  
 – **Usage**  
   \* Visits a ComplementExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ConditionalExpression node )  
 – **Usage**  
   \* Visits a ConditionalExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ConstructorDeclaration node )  
 – **Usage**  
   \* Visits a ConstructorDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ConstructorInvocation node )  
 – **Usage**  
   \* Visits a ConstructorInvocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ContinueStatement node )

- **Usage**
  - \* Visits a ContinueStatement
- **Parameters**
  - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
```

  - **Usage**
    - \* Visits an DivideAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideExpression node )
```

  - **Usage**
    - \* Visits a DivideExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DoStatement node )
```

  - **Usage**
    - \* Visits a DoStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.EmptyStatement node )
```

  - **Usage**
    - \* Visits an EmptyStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.EqualExpression node )
```

  - **Usage**
    - \* Visits a EqualExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
```

  - **Usage**

- \* Visits a ExclusiveOrAssignExpression
  - **Parameters**
  - \* **node** - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
  - **Usage**
  - \* Visits a ExclusiveOrExpression
  - **Parameters**
  - \* **node** - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.FieldDeclaration node )
  - **Usage**
  - \* Visits a FieldDeclaration
  - **Parameters**
  - \* **node** - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.FormalParameter node )
  - **Usage**
  - \* Visits a FormalParameter
  - **Parameters**
  - \* **node** - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.ForStatement node )
  - **Usage**
  - \* Visits a ForStatement
  - **Parameters**
  - \* **node** - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.FunctionCall node )
  - **Usage**
  - \* Visits a FunctionCall
  - **Parameters**
  - \* **node** - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.GreaterExpression node )
  - **Usage**
  - \* Visits a GreaterExpression
  - **Parameters**

- 
- \* node - the node to visit

---

    - *visit*

public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )

      - Usage
      - \* Visits a GreaterOrEqualExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )

      - Usage
      - \* Visits a IfThenElseStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.IfThenStatement node )

      - Usage
      - \* Visits a IfThenStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ImportDeclaration node )

      - Usage
      - \* Visits an ImportDeclaration
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.InnerAllocation node )

      - Usage
      - \* Visits an InnerAllocation
      - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )

      - Usage
      - \* Visits an InnerClassAllocation
      - Parameters
      - \* node - the node to visit

---

- *visit*  
 public Object **visit**( koala.dynamicjava.tree.InstanceInitializer node )  
 – **Usage**  
   \* Visits a InstanceInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.InstanceOfExpression node )  
 – **Usage**  
   \* Visits an InstanceOfExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.InterfaceDeclaration node )  
 – **Usage**  
   \* Visits a InterfaceDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.LabeledStatement node )  
 – **Usage**  
   \* Visits a LabeledStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.LessExpression node )  
 – **Usage**  
   \* Visits a LessExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.LessOrEqualExpression node )  
 – **Usage**  
   \* Visits a LessOrEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.Literal node )



- **Usage**
  - \* Visits a Literal
- **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
```

  - **Usage**
    - \* Visits a MethodDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MinusExpression node )
```

  - **Usage**
    - \* Visits a MinusExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

  - **Usage**
    - \* Visits an MultiplyAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

  - **Usage**
    - \* Visits a MultiplyExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

  - **Usage**
    - \* Visits a NotEqualExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotExpression node )
```

  - **Usage**
    - \* Visits a NotExpression

- **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
```

    - **Usage**
      - \* Visits a ObjectFieldAccess
    - **Parameters**
      - \* **node** - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
```

      - **Usage**
        - \* Visits a ObjectMethodCall
      - **Parameters**
        - \* **node** - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.OrExpression node )
```

        - **Usage**
          - \* Visits a OrExpression
        - **Parameters**
          - \* **node** - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
```

          - **Usage**
            - \* Visits an PackageDeclaration
          - **Parameters**
            - \* **node** - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.PlusExpression node )
```

            - **Usage**
              - \* Visits a PlusExpression
            - **Parameters**
              - \* **node** - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement node )
```

              - **Usage**
                - \* Visits a PostDecrement
              - **Parameters**
                - \* **node** - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.PostIncrement node )
```

  - **Usage**  
 \* Visits a PostIncrement
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.PreDecrement node )
```

  - **Usage**  
 \* Visits a PreDecrement
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.PreIncrement node )
```

  - **Usage**  
 \* Visits a PreIncrement
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
```

  - **Usage**  
 \* Visits a PrimitiveType
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.QualifiedName node )
```

  - **Usage**  
 \* Visits a QualifiedName
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ReferenceType node )
```

  - **Usage**  
 \* Visits a ReferenceType
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
```

- **Usage**
    - \* Visits an RemainderAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderExpression  node )
```

  - **Usage**
    - \* Visits a RemainderExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReturnStatement  node )
```

  - **Usage**
    - \* Visits a ReturnStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression  node )
```

  - **Usage**
    - \* Visits an ShiftLeftAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftExpression  node )
```

  - **Usage**
    - \* Visits a ShiftLeftExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression  node )
```

  - **Usage**
    - \* Visits an ShiftRightAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightExpression  node )
```

  - **Usage**

- \* Visits a ShiftRightExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
```

    - **Usage**
      - \* Visits an SimpleAllocation
    - **Parameters**
      - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )
```

    - **Usage**
      - \* Visits an SimpleAssignExpression
    - **Parameters**
      - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )
```

    - **Usage**
      - \* Visits a StaticFieldAccess
    - **Parameters**
      - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.StaticMethodCall node )
```

    - **Usage**
      - \* Visits a StaticMethodCall
    - **Parameters**
      - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
```

    - **Usage**
      - \* Visits an SubtractAssignExpression
    - **Parameters**
      - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractExpression node )
```

    - **Usage**
      - \* Visits a SubtractExpression
    - **Parameters**

- 
- \* node - the node to visit

---

    - *visit*

public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )

      - Usage
        - \* Visits a SuperFieldAccess
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.SuperMethodCall node )

      - Usage
        - \* Visits a SuperMethodCall
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.SwitchBlock node )

      - Usage
        - \* Visits a SwitchBlock
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.SwitchStatement node )

      - Usage
        - \* Visits a SwitchStatement
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )

      - Usage
        - \* Visits a SynchronizedStatement
      - Parameters
        - \* node - the node to visit

---
    - *visit*

public Object visit( koala.dynamicjava.tree.ThisExpression node )

      - Usage
        - \* Visits a ThisExpression
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ThrowStatement node )  
 – Usage  
   \* Visits a ThrowStatement  
 – Parameters  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.TryStatement node )  
 – Usage  
   \* Visits a TryStatement  
 – Parameters  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.TypeExpression node )  
 – Usage  
   \* Visits a TypeExpression  
 – Parameters  
   \* node - the node to visit  


---
- *visit*  
 public Object visit(  
 koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )  
 – Usage  
   \* Visits an UnsignedShiftRightAssignExpression  
 – Parameters  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression  
 node )  
 – Usage  
   \* Visits a UnsignedShiftRightExpression  
 – Parameters  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.VariableDeclaration node )  
 – Usage  
   \* Visits a VariableDeclaration  
 – Parameters  
   \* node - the node to visit  


---

- *visit*

```
public Object visit( koala.dynamicjava.tree.WhileStatement  node )
```

- **Usage**

- \* Visits a WhileStatement

- **Parameters**

- \* **node** - the node to visit



## Chapter 6

# Package jeliot.gui

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Classes</b>	
<b>AboutWindow</b> .....	177
<i>The design of the about window</i>	
<b>CodeEditor</b> .....	199
<i>The simple code editor for the users to code their algorithm.</i>	
<b>CodePane</b> .....	228
<i>This is the component that shows and highlights the program while Jeliot is animating, called also code view.</i>	
<b>DraggableComponent</b> .....	252
<i>Not currently used in Jeliot 3.</i>	
<b>HelpWindow</b> .....	277
<i>The design of the help window.</i>	
<b>JavaFileFilter</b> .....	299
<i>Filter for the file chooser to only show the java source code files.</i>	
<b>JeliotWindow</b> .....	300
<i>The main window of the Jeliot 3.</i>	
<b>LineNumbers</b> .....	308
<i>The LineNumbers component is used to show the line numbers in the scroll panes left side in the code view and code editor.</i>	
<b>LoadJeliot</b> .....	332
<i>This class is not used in the current version of Jeliot.</i>	
<b>OutputConsole</b> .....	333
<i>OutputConsole is a text area on which the output of a user's program is printed.</i>	
<b>TheaterPopup</b> .....	362
<i>Currently this class is not used in the Jeliot.</i>	

---

## 6.1 Classes

### 6.1.1 CLASS AboutWindow

---

The design of the about window

#### DECLARATION

---

```
public class AboutWindow
extends javax.swing.JFrame
```

#### SERIALIZABLE FIELDS

---

- private JEditorPane aboutPane
  - The pane where info about the program and Licensing will be shown.
- private JScrollPane jsp
  - The pane that handles the scrolling of the editor pane showing the content.

#### FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for gui package
- private JEditorPane aboutPane
  - The pane where info about the program and Licensing will be shown.
- private JScrollPane jsp
  - The pane that handles the scrolling of the editor pane showing the content.

#### CONSTRUCTORS

---

- *AboutWindow*

```
public AboutWindow( java.awt.Image  icon, java.lang.String  udir )
```

  - **Usage**
    - \* constructs the AboutWindow by creating a JFrame. Sets inside the JFrame JScrollPane with JEditorPane editorPane. Sets the size of the JFrame as 400 x 600
  - **Parameters**
    - \* **icon** - Icon to be shown in the upper right corner of the window.
    - \* **udir** - directory of the current invocation

METHODS

---

- *showURL*  
`public void showURL( java.net.URL url )`
  - **Usage**
    - \* Shows the given url in the editor pane.
  - **Parameters**
    - \* `url` - The document in the url will be showed in JEditorPane editorPane.

METHODS INHERITED FROM CLASS `javax.swing.JFrame`

---

- *<clinit>*  
`static void <clinit>( )`
- *addImpl*  
`protected void addImpl( java.awt.Component , java.lang.Object , int )`
- *createRootPane*  
`protected JRootPane createRootPane( )`
- *createRootPaneException*  
`private Error createRootPaneException( java.lang.String )`
- *frameInit*  
`protected void frameInit( )`
- *getAccessibleContext*  
`public AccessibleContext getAccessibleContext( )`
- *getContentPane*  
`public Container getContentPane( )`
- *getDefaultCloseOperation*  
`public int getDefaultCloseOperation( )`
- *getGlassPane*  
`public Component getGlassPane( )`
- *getJMenuBar*  
`public JMenuBar getJMenuBar( )`
- *getLayeredPane*  
`public JLayeredPane getLayeredPane( )`
- *getRootPane*  
`public JRootPane getRootPane( )`
- *isDefaultLookAndFeelDecorated*  
`public static boolean isDefaultLookAndFeelDecorated( )`
- *isRootPaneCheckingEnabled*  
`protected boolean isRootPaneCheckingEnabled( )`
- *paramString*  
`protected String paramString( )`
- *processWindowEvent*  
`protected void processWindowEvent( java.awt.event.WindowEvent )`
- *remove*  
`public void remove( java.awt.Component )`
- *setContentPane*  
`public void setContentPane( java.awt.Container )`

- *setDefaultCloseOperation*  
public void setDefaultCloseOperation( int )
- *setDefaultLookAndFeelDecorated*  
public static void setDefaultLookAndFeelDecorated( boolean )
- *setGlassPane*  
public void setGlassPane( java.awt.Component )
- *setJMenuBar*  
public void setJMenuBar( javax.swing.JMenuBar )
- *setLayeredPane*  
public void setLayeredPane( javax.swing.JLayeredPane )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setRootPane*  
protected void setRootPane( javax.swing.JRootPane )
- *setRootPaneCheckingEnabled*  
protected void setRootPaneCheckingEnabled( boolean )
- *update*  
public void update( java.awt.Graphics )

#### METHODS INHERITED FROM CLASS java.awt.Frame

---

- *<clinit>*  
static void <clinit>( )
- *addNotify*  
public void addNotify( )
- *addToFrameList*  
void addToFrameList( )
- *constructComponentName*  
String constructComponentName( )
- *finalize*  
protected void finalize( )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getCursorType*  
public int getCursorType( )
- *getExtendedState*  
public synchronized int getExtendedState( )
- *getFrames*  
public static Frame getFrames( )
- *getIconImage*  
public Image getIconImage( )
- *getMaximizedBounds*  
public Rectangle getMaximizedBounds( )
- *getMenuBar*  
public MenuBar getMenuBar( )
- *getState*  
public synchronized int getState( )
- *getTitle*  
public String getTitle( )

- *init*  
private void init( java.lang.String , java.awt.GraphicsConfiguration )
- *initIDs*  
private static native void initIDs( )
- *isResizable*  
public boolean isResizable( )
- *isUndecorated*  
public boolean isUndecorated( )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public void remove( java.awt.MenuComponent )
- *removeFromFrameList*  
void removeFromFrameList( )
- *removeNotify*  
public void removeNotify( )
- *setCursor*  
public void setCursor( int )
- *setExtendedState*  
public synchronized void setExtendedState( int )
- *setIconImage*  
public synchronized void setIconImage( java.awt.Image )
- *setMaximizedBounds*  
public synchronized void setMaximizedBounds( java.awt.Rectangle )
- *setMenuBar*  
public void setMenuBar( java.awt.MenuBar )
- *setResizable*  
public void setResizable( boolean )
- *setState*  
public synchronized void setState( int )
- *setTitle*  
public void setTitle( java.lang.String )
- *setUndecorated*  
public void setUndecorated( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

METHODS INHERITED FROM CLASS `java.awt.Window`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *addNotify*  
`public void addNotify( )`
  - *addOwnedWindow*  
`void addOwnedWindow( java.lang.ref.WeakReference )`
  - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )`
  - *addWindowFocusListener*  
`public synchronized void addWindowFocusListener(  
java.awt.event.WindowFocusListener )`
  - *addWindowListener*  
`public synchronized void addWindowListener( java.awt.event.WindowListener )`
  - *addWindowStateListener*  
`public synchronized void addWindowStateListener(  
java.awt.event.WindowStateListener )`
  - *adjustDecendantsOnParent*  
`void adjustDecendantsOnParent( int )`
  - *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
  - *applyResourceBundle*  
`public void applyResourceBundle( java.util.ResourceBundle )`
  - *applyResourceBundle*  
`public void applyResourceBundle( java.lang.String )`
  - *clearMostRecentFocusOwnerOnHide*  
`final void clearMostRecentFocusOwnerOnHide( )`
  - *connectOwnedWindow*  
`void connectOwnedWindow( java.awt.Window )`
  - *constructComponentName*  
`String constructComponentName( )`
  - *createBufferStrategy*  
`public void createBufferStrategy( int )`
  - *createBufferStrategy*  
`public void createBufferStrategy( int , java.awt.BufferCapabilities )`
  - *deliverMouseWheelToAncestor*  
`void deliverMouseWheelToAncestor( java.awt.event.MouseWheelEvent )`
  - *dispatchEventImpl*  
`void dispatchEventImpl( java.awt.AWTEvent )`
  - *dispatchMouseWheelToAncestor*  
`boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )`
  - *dispose*  
`public void dispose( )`
  - *eventEnabled*  
`boolean eventEnabled( java.awt.AWTEvent )`
  - *finalize*  
`protected void finalize( )`

- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getBufferStrategy*  
public BufferStrategy getBufferStrategy( )
- *getFocusableWindowState*  
public boolean getFocusableWindowState( )
- *getFocusCycleRootAncestor*  
public final Container getFocusCycleRootAncestor( )
- *getFocusOwner*  
public Component getFocusOwner( )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getInputContext*  
public InputContext getInputContext( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocale*  
public Locale getLocale( )
- *getMostRecentFocusOwner*  
public Component getMostRecentFocusOwner( )
- *getOwnedWindows*  
public Window getOwnedWindows( )
- *getOwner*  
public Window getOwner( )
- *getTemporaryLostComponent*  
Component getTemporaryLostComponent( )
- *getToolkit*  
public Toolkit getToolkit( )
- *getWarningString*  
public final String getWarningString( )
- *getWindowFocusListeners*  
public synchronized WindowFocusListener getWindowFocusListeners( )
- *getWindowListeners*  
public synchronized WindowListener getWindowListeners( )
- *getWindowStateListeners*  
public synchronized WindowStateListener getWindowStateListeners( )
- *hide*  
public void hide( )
- *init*  
private void init( java.awt.GraphicsConfiguration )
- *initIDs*  
private static native void initIDs( )
- *isActive*  
public boolean isActive( )
- *isFocusableWindow*  
public final boolean isFocusableWindow( )
- *isFocusCycleRoot*  
public final boolean isFocusCycleRoot( )

- *isFocused*  
public boolean isFocused( )
- *isShowing*  
public boolean isShowing( )
- *ownedInit*  
private void ownedInit( java.awt.Window )
- *pack*  
public void pack( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postWindowEvent*  
synchronized void postWindowEvent( int )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processWindowEvent*  
protected void processWindowEvent( java.awt.event.WindowEvent )
- *processWindowFocusEvent*  
protected void processWindowFocusEvent( java.awt.event.WindowEvent )
- *processWindowStateEvent*  
protected void processWindowStateEvent( java.awt.event.WindowEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removeOwnedWindow*  
void removeOwnedWindow( java.lang.ref.WeakReference )
- *removeWindowFocusListener*  
public synchronized void removeWindowFocusListener( java.awt.event.WindowFocusListener )
- *removeWindowListener*  
public synchronized void removeWindowListener( java.awt.event.WindowListener )
- *removeWindowStateListener*  
public synchronized void removeWindowStateListener( java.awt.event.WindowStateListener )
- *resetGC*  
void resetGC( )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setFocusableWindowState*  
public void setFocusableWindowState( boolean )
- *setFocusCycleRoot*  
public final void setFocusCycleRoot( boolean )
- *setLocationRelativeTo*  
public void setLocationRelativeTo( java.awt.Component )
- *setTemporaryLostComponent*  
Component setTemporaryLostComponent( java.awt.Component )
- *setWarningString*  
private void setWarningString( )



- *show*  
public void show( )
- *toBack*  
public void toBack( )
- *ToFront*  
public void toFront( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

---

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int )
- *adjustDescendants*  
void adjustDescendants( int )
- *adjustListeningChildren*  
void adjustListeningChildren( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *checkGD*  
void checkGD( java.lang.String )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )

- *containsFocus*  
final boolean containsFocus( )
- *countComponents*  
public int countComponents( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )

- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )
- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseEventTarget*  
Component getMouseEventTarget( int , int , boolean )
- *getMouseEventTarget*  
private Component getMouseEventTarget( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component getMouseEventTargetImpl( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *insets*  
public Insets insets( )
- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )

- *lightweightPaint*  
void **lightweightPaint**( java.awt.Graphics )
- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics )
- *list*  
public void **list**( java.io.PrintStream , int )
- *list*  
public void **list**( java.io.PrintWriter , int )
- *locate*  
public Component **locate**( int , int )
- *minimumSize*  
public Dimension **minimumSize**( )
- *nextFocusHelper*  
boolean **nextFocusHelper**( )
- *numListening*  
int **numListening**( long )
- *paint*  
public void **paint**( java.awt.Graphics )
- *paintComponents*  
public void **paintComponents**( java.awt.Graphics )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics )
- *paramString*  
protected String **paramString**( )
- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *preProcessKeyEvent*  
void **preProcessKeyEvent**( java.awt.event.KeyEvent )
- *print*  
public void **print**( java.awt.Graphics )
- *printComponents*  
public void **printComponents**( java.awt.Graphics )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *remove*  
public void **remove**( java.awt.Component )
- *remove*  
public void **remove**( int )

- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void setFocusTraversalPolicy( java.awt.FocusTraversalPolicy )
- *setFont*  
public void setFont( java.awt.Font )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setZOrder*  
void setZOrder( java.awt.Component , int )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusDownCycle*  
public void transferFocusDownCycle( )
- *update*  
public void update( java.awt.Graphics )
- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )

- *addInputMethodListener*  
public synchronized void addInputMethodListener(  
java.awt.event.InputMethodListener    )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener    )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener    )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener(  
java.awt.event.MouseMotionListener    )
- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener  
)
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long    , int    )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation    )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int    )
- *areInputMethodsEnabled*  
boolean areInputMethodsEnabled( )
- *autoProcessMouseWheel*  
void autoProcessMouseWheel( java.awt.event.MouseWheelEvent    )
- *autoTransferFocus*  
final void autoTransferFocus( boolean    )
- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String    )
- *checkImage*  
public int checkImage( java.awt.Image    , java.awt.image.ImageObserver    )
- *checkImage*  
public int checkImage( java.awt.Image    , int    , int    , java.awt.image.ImageObserver  
)
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent    , java.awt.AWTEvent    )
- *constructComponentName*  
String constructComponentName( )

- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *doAutoTransfer*  
private void doAutoTransfer( boolean )
- *doLayout*  
public void doLayout( )
- *enable*  
public void enable( )
- *enable*  
public void enable( boolean )
- *enableEvents*  
protected final void enableEvents( long )
- *enableInputMethods*  
public void enableInputMethods( boolean )

- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent( )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getBackBuffer*  
Image getBackBuffer( )
- *getBackground*  
public Color getBackground( )
- *getBounds*  
public Rectangle getBounds( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy getBufferStrategy( )
- *getColorModel*  
public ColorModel getColorModel( )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener getComponentListeners( )
- *getComponentOrientation*  
public ComponentOrientation getComponentOrientation( )
- *getCursor*  
public Cursor getCursor( )
- *getDropTarget*  
public synchronized DropTarget getDropTarget( )
- *getFocusCycleRootAncestor*  
public Container getFocusCycleRootAncestor( )
- *getFocusListeners*  
public synchronized FocusListener getFocusListeners( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set getFocusTraversalKeys\_NoIDCheck( int )



- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int    )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font    )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class    )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point    )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )

- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )
- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )
- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )

- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )
- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )

- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics )
- *list*  
public void **list**( )
- *list*  
public void **list**( java.io.PrintStream )
- *list*  
public void **list**( java.io.PrintStream , int )
- *list*  
public void **list**( java.io.PrintWriter )
- *list*  
public void **list**( java.io.PrintWriter , int )
- *locate*  
public Component **locate**( int , int )
- *location*  
public Point **location**( )
- *lostFocus*  
public boolean **lostFocus**( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension **minimumSize**( )
- *mouseDown*  
public boolean **mouseDown**( java.awt.Event , int , int )
- *mouseDrag*  
public boolean **mouseDrag**( java.awt.Event , int , int )
- *mouseEnter*  
public boolean **mouseEnter**( java.awt.Event , int , int )
- *mouseExit*  
public boolean **mouseExit**( java.awt.Event , int , int )
- *mouseMove*  
public boolean **mouseMove**( java.awt.Event , int , int )
- *mouseUp*  
public boolean **mouseUp**( java.awt.Event , int , int )
- *move*  
public void **move**( int , int )
- *nextFocus*  
public void **nextFocus**( )
- *nextFocusHelper*  
boolean **nextFocusHelper**( )
- *numListening*  
int **numListening**( long )
- *paint*  
public void **paint**( java.awt.Graphics )
- *paintAll*  
public void **paintAll**( java.awt.Graphics )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics )
- *paramString*  
protected String **paramString**( )
- *postEvent*  
public boolean **postEvent**( java.awt.Event )

- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *prepareImage*  
public boolean **prepareImage**( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean **prepareImage**( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void **print**( java.awt.Graphics )
- *printAll*  
public void **printAll**( java.awt.Graphics )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics )
- *processComponentEvent*  
protected void **processComponentEvent**( java.awt.event.ComponentEvent )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent )
- *processFocusEvent*  
protected void **processFocusEvent**( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void **processHierarchyBoundsEvent**( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void **processHierarchyEvent**( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void **processInputMethodEvent**( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void **processKeyEvent**( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void **processMouseEvent**( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void **processMouseMotionEvent**( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void **processMouseWheelEvent**( java.awt.event.MouseWheelEvent )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *remove*  
public synchronized void **remove**( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void **removeComponentListener**( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void **removeFocusListener**( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void **removeHierarchyBoundsListener**( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void **removeHierarchyListener**( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void **removeInputMethodListener**( java.awt.event.InputMethodListener )

- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *repaint*  
public void repaint( )
- *repaint*  
public void repaint( int , int , int , int )
- *repaint*  
public void repaint( long )
- *repaint*  
public void repaint( long , int , int , int , int )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )

- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )

- *transferFocusUpCycle*  
public void **transferFocusUpCycle**( )
- *update*  
public void **update**( java.awt.Graphics )
- *updateCursorImmediately*  
final void **updateCursorImmediately**( )
- *validate*  
public void **validate**( )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 6.1.2 CLASS CodeEditor

---

The simple code editor for the users to code their algorithm.

#### DECLARATION

---

```
public class CodeEditor
extends javax.swing.JComponent
```

#### SERIALIZABLE FIELDS

---

- private String template
  - The String for the basic code template that is shown to the user in the beginning.
- private Font areaFont
  - Font for the editor area.
- private Insets insets
  - Insets for the text. Used for the layout.
- private LineNumbers nb
  - Line numbering component that handles the correct line numbering in the editor view.
- private boolean changed
  - Tells whether or not the current file is changed since last loading or saving. Used to determine when the saving dialog should be popped up.
- private File currentFile
  - Pointing to the current file that is edited for saving before the compilation.
- private DocumentListener dcl
  - Document listener is used to handle the line numbering correctly when the new line is created or the document is scrolled by the user
- private JFileChooser fileChooser
  - The file chooser in which the users can load and save the program codes.



- private JFrame masterFrame
  - The master frame.
- private ActionListener saver
  - ActionListener that handles the saving of the program code from the code area.
- private ActionListener loader
  - ActionListener that handles the loading of the program code to the code area.
- private ActionListener clearer
  - ActionListener that handles the clearing of the code area.
- private ActionListener cutter
  - ActionListener that handles the clearing of the code area.
- private ActionListener copyist
  - ActionListener that handles the copying of the code area.
- private ActionListener pasteur
  - ActionListener that handles the pasting of the code area.
- private ActionListener allSelector
  - ActionListener that handles the selection of the whole code area.

## FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for gui package
- private String template
  - The String for the basic code template that is shown to the user in the beginning.
- private Font areaFont
  - Font for the editor area.
- private Insets insets
  - Insets for the text. Used for the layout.
- private LineNumbers nb
  - Line numbering component that handles the correct line numbering in the editor view.
- private boolean changed
  - Tells whether or not the current file is changed since last loading or saving. Used to determine when the saving dialog should be popped up.
- private File currentFile
  - Pointing to the current file that is edited for saving before the compilation.
- private DocumentListener dcl

- Document listener is used to handle the line numbering correctly when the new line is created or the document is scrolled by the user
- private JFileChooser fileChooser
  - The file chooser in which the users can load and save the program codes.
- private JFrame masterFrame
  - The master frame.
- private ActionListener saver
  - ActionListener that handles the saving of the program code from the code area.
- private ActionListener loader
  - ActionListener that handles the loading of the program code to the code area.
- private ActionListener clearer
  - ActionListener that handles the clearing of the code area.
- private ActionListener cutter
  - ActionListener that handles the clearing of the code area.
- private ActionListener copyist
  - ActionListener that handles the copying of the code area.
- private ActionListener pasteur
  - ActionListener that handles the pasting of the code area.
- private ActionListener allSelector
  - ActionListener that handles the selection of the whole code area.

## CONSTRUCTORS

---

- *CodeEditor*  
**public CodeEditor( )**
  - **Usage**
    - \* Sets the layout and adds the JScrollPane with JTextArea area and JToolBar in it. Initializes the FileChooser.

## METHODS

---

- *calculateLines*  
**public int calculateLines( java.lang.String text )**
  - **Usage**
    - \* Calculates the number of lines in the program source code.
  - **Parameters**
    - \* **text** - the program source code.

- **Returns** - the number of lines in the given program source code

---

- *clearProgram*

**void clearProgram( )**

- **Usage**

\* Sets in JTextAre area the default text as given in template.

---

- *getCurrentFile*

**public File getCurrentFile( )**

- **Usage**

\* Returns the current file that is loaded.

- **Returns** - The File object pointing to the current file.

---

- *getProgram*

**public String getProgram( )**

- **Usage**

\* Method returns the program code inside the JTextArea as String -object  
Tabulators are changed to spaces for uniform handling of white spaces. One  
tabulator corresponds four ASCII white spaces.

- **Returns** - The program code inside the JTextArea area.

---

- *highlight*

**public void highlight( jeliot.theater.Highlight h )**

- **Usage**

\* Method highlights the specified code area by selecting it.

- **Parameters**

\* **h** - contains the area that should be highlighted.

---

- *highlightStatement*

**public void highlightStatement( jeliot.theater.Highlight h )**

- **Usage**

\* Method highlights the specified Statement area by selecting it.

- **Parameters**

\* **h** - contains the area that should be highlighted.

---

- *initFileChooser*

**private void initFileChooser( )**

- **Usage**

\* Sets up the file chooser with the user's working directory as default directory.

---

- *isChanged*

**public boolean isChanged( )**

- **Usage**

\* returns true if the document is changed and false if it is not changed. This is the  
value of the changed field.

- **Returns** - if the document is changed or not.

---

- *loadProgram*

**void loadProgram( )**

- **Usage**

- \* Loads the program from a file to the JTextArea area. Uses readProgram(File file) method to read the file. Uses setProgram(String str) method to set the content of the file into the JTextArea area.

- **See Also**

- \* jeliot.gui.CodeEditor.readProgram(File)
  - \* jeliot.gui.CodeEditor.setProgram(String)
- 

- *makeEditMenu*

**JMenu makeEditMenu( )**

- **Usage**

- \* Constructs the Edit menu.

- **Returns** - The Edit menu

---

- *makeProgramMenu*

**JMenu makeProgramMenu( )**

- **Usage**

- \* Constructs the Program menu.

- **Returns** - The Program menu

---

- *makeScrollPane*

**public JComponent makeScrollPane( )**

- **Usage**

- \* Creates a ScrollPane object with a LineNumbers set as its left side.

- **Returns** - a scroll pane with certain parameters and line numbering

- **See Also**

- \* jeliot.gui.LineNumbers ( in 6.1.8, page 308)
- 

- *makeToolBar*

**private JToolBar makeToolBar( )**

- **Usage**

- \* The method makes the Buttons for the toolbar of the codearea. Then it adds the button to the JToolBar and returns it. Uses makeToolButton(String, String, ActionListener) -method.

- **Returns** - The finished toolbar for the code editor.

- **See Also**

- \* jeliot.gui.CodeEditor.makeToolButton(String, String, ActionListener)
- 

- *makeToolButton*

**private JButton makeToolButton( java.lang.String label, java.lang.String iconName, java.awt.event.ActionListener listener )**

- **Usage**
    - \* Makes the JButton from the parameters given.
  - **Parameters**
    - \* **label** - The label of the button.
    - \* **iconName** - The name of the image for the button.
    - \* **listener** - The actionlistener for that button.
  - **Returns** - The constructed button from the given parameters.
- 

- *readProgram*

**String readProgram( java.io.File file )**

- **Usage**
    - \* Reads the content of the given file and returns the content of the file as String.
  - **Parameters**
    - \* **file** - The file from which the content is read and returned for the use of loadProgram() method.
  - **Returns** - The content of the file that was given as parameter.
  - **See Also**
    - \* jeliot.gui.CodeEditor.loadProgram()
- 

- *saveProgram*

**void saveProgram( )**

- **Usage**
    - \* Saves the program from the JTextArea area to the file. Uses writeProgram(File file) method to write the code into a file.
  - **See Also**
    - \* jeliot.gui.CodeEditor.writeProgram(File)
- 

- *setChanged*

**public void setChanged( boolean changed )**

- **Usage**
    - \* Set wheter or not the document is changed or not.
  - **Parameters**
    - \* **changed** - if true the document is changed if false the document is not changed (means that it is just loaded or saved).
- 

- *setMasterFrame*

**public void setMasterFrame( javax.swing.JFrame frame )**

- **Usage**
    - \* Set the given frame as the masterFrame.
  - **Parameters**
    - \* **frame** - The Frame that is set as new masterFrame.
- 

- *setProgram*

**void setProgram( java.lang.String program )**

- **Usage**

\* The given String program object will be set as the text inside the JTextArea area.

– **Parameters**

\* **program** - The string that will be set in JTextArea area as the program code.

---

• *validateScrollPane*

**public void validateScrollPane( )**

– **Usage**

\* Validates the scroll pane's line numbering.

---

• *writeProgram*

**public void writeProgram( java.io.File file )**

– **Usage**

\* Saves the content of the JTextArea area to a given file.

– **Parameters**

\* **file** - The file where the content of JTextArea is saved.

---

#### METHODS INHERITED FROM CLASS javax.swing.JComponent

---

• *\_paintImmediately*

**void \_paintImmediately( int , int , int , int )**

• *<clinit>*

**static void <clinit>( )**

• *addAncestorListener*

**public void addAncestorListener( javax.swing.event.AncestorListener )**

• *addNotify*

**public void addNotify( )**

• *addPropertyChangeListener*

**public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )**

• *addPropertyChangeListener*

**public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )**

• *addVetoableChangeListener*

**public synchronized void addVetoableChangeListener( java.beans.VetoableChangeListener )**

• *adjustPaintFlags*

**private void adjustPaintFlags( )**

• *alwaysOnTop*

**boolean alwaysOnTop( )**

• *checkIfChildObscuredBySibling*

**boolean checkIfChildObscuredBySibling( )**

• *componentInputMapChanged*

**void componentInputMapChanged( javax.swing.ComponentInputMap )**

• *computeVisibleRect*

**static final void computeVisibleRect( java.awt.Component , java.awt.Rectangle )**

• *computeVisibleRect*

**public void computeVisibleRect( java.awt.Rectangle )**

- *compWriteObjectNotify*  
void **compWriteObjectNotify**( )
- *contains*  
public boolean **contains**( int , int )
- *createToolTip*  
public JToolTip **createToolTip**( )
- *deregisterNextFocusableComponent*  
private void **deregisterNextFocusableComponent**( )
- *disable*  
public void **disable**( )
- *enable*  
public void **enable**( )
- *enableSerialization*  
void **enableSerialization**( )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , boolean , boolean )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , byte , byte )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , char , char )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , double , double )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , float , float )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , int , int )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , long , long )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , short , short )
- *fireVetoableChange*  
protected void **fireVetoableChange**( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getActionForKeyStroke*  
public ActionListener **getActionForKeyStroke**( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap **getActionMap**( )
- *getActionMap*  
final ActionMap **getActionMap**( boolean )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getAncestorListeners*  
public AncestorListener **getAncestorListeners**( )

- *getAutoscrolls*  
public boolean getAutoscrolls( )
- *getBorder*  
public Border getBorder( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary getClientProperties( )
- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )
- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )
- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )



- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )
- *getToolTipText*  
public String getToolTipText( )
- *getToolTipText*  
public String getToolTipText( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container getTopLevelAncestor( )
- *getTransferHandler*  
public TransferHandler getTransferHandler( )
- *getUIClassID*  
public String getUIClassID( )
- *getVerifyInputWhenFocusTarget*  
public boolean getVerifyInputWhenFocusTarget( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener getVetoableChangeListeners( )
- *getWidth*  
public int getWidth( )
- *getVisibleRect*  
public Rectangle getVisibleRect( )
- *getWriteObjCounter*  
static byte getWriteObjCounter( javax.swing.JComponent )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *grabFocus*  
public void grabFocus( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component )

- *isManagingFocus*  
public boolean isManagingFocus( )
- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )
- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )

- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener , javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener , java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener( java.beans.VetoableChangeListener )
- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )

- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )
- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )

- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )
- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

METHODS INHERITED FROM CLASS `java.awt.Container`

- *<clinit>*  
`static void <clinit>( )`
- *add*  
`public Component add( java.awt.Component )`
- *add*  
`public Component add( java.awt.Component , int )`
- *add*  
`public void add( java.awt.Component , java.lang.Object )`
- *add*  
`public void add( java.awt.Component , java.lang.Object , int )`
- *add*  
`public Component add( java.lang.String , java.awt.Component )`
- *addContainerListener*  
`public synchronized void addContainerListener( java.awt.event.ContainerListener )`
- *addImpl*  
`protected void addImpl( java.awt.Component , java.lang.Object , int )`
- *addNotify*  
`public void addNotify( )`
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener )`
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
- *adjustDecendantsOnParent*  
`void adjustDecendantsOnParent( int )`
- *adjustDescendants*  
`void adjustDescendants( int )`
- *adjustListeningChildren*  
`void adjustListeningChildren( long , int )`
- *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
- *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
- *checkGD*  
`void checkGD( java.lang.String )`
- *clearCurrentFocusCycleRootOnHide*  
`void clearCurrentFocusCycleRootOnHide( )`
- *clearMostRecentFocusOwnerOnHide*  
`void clearMostRecentFocusOwnerOnHide( )`
- *containsFocus*  
`final boolean containsFocus( )`
- *countComponents*  
`public int countComponents( )`
- *countHierarchyMembers*  
`int countHierarchyMembers( )`
- *createChildHierarchyEvents*  
`void createChildHierarchyEvents( int , long , boolean )`

- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )

- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseEventTarget*  
Component getMouseEventTarget( int , int , boolean )
- *getMouseEventTarget*  
private Component getMouseEventTarget( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component getMouseEventTargetImpl( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *insets*  
public Insets insets( )
- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )



- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )
- *printComponents*  
public void printComponents( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processContainerEvent*  
protected void processContainerEvent( java.awt.event.ContainerEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *proxyEnableEvents*  
void proxyEnableEvents( long )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public void remove( java.awt.Component )
- *remove*  
public void remove( int )
- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )

- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void setFocusTraversalPolicy( java.awt.FocusTraversalPolicy )
- *setFont*  
public void setFont( java.awt.Font )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setZOrder*  
void setZOrder( java.awt.Component , int )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusDownCycle*  
public void transferFocusDownCycle( )
- *update*  
public void update( java.awt.Graphics )
- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )
- *addInputMethodListener*  
public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )

- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *areInputMethodsEnabled*  
boolean areInputMethodsEnabled( )
- *autoProcessMouseWheel*  
void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )
- *autoTransferFocus*  
final void autoTransferFocus( boolean )
- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )

- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *doAutoTransfer*  
private void doAutoTransfer( boolean )
- *doLayout*  
public void doLayout( )
- *enable*  
public void enable( )
- *enable*  
public void enable( boolean )
- *enableEvents*  
protected final void enableEvents( long )
- *enableInputMethods*  
public void enableInputMethods( boolean )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , int , int )

- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent( )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getBackBuffer*  
Image getBackBuffer( )
- *getBackground*  
public Color getBackground( )
- *getBounds*  
public Rectangle getBounds( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy getBufferStrategy( )
- *getColorModel*  
public ColorModel getColorModel( )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener getComponentListeners( )
- *getComponentOrientation*  
public ComponentOrientation getComponentOrientation( )
- *getCursor*  
public Cursor getCursor( )
- *getDropTarget*  
public synchronized DropTarget getDropTarget( )
- *getFocusCycleRootAncestor*  
public Container getFocusCycleRootAncestor( )
- *getFocusListeners*  
public synchronized FocusListener getFocusListeners( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set getFocusTraversalKeys\_NoIDCheck( int )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalKeysEnabled*  
public boolean getFocusTraversalKeysEnabled( )
- *getFont\_NoClientCode*  
final Font getFont\_NoClientCode( )
- *getFont*  
public Font getFont( )

- *getFontMetrics*  
public FontMetrics getFontMetrics( java.awt.Font    )
- *getForeground*  
public Color getForeground(    )
- *getGraphics*  
public Graphics getGraphics(    )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration(    )
- *getHeight*  
public int getHeight(    )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener getHierarchyBoundsListeners(    )
- *getHierarchyListeners*  
public synchronized HierarchyListener getHierarchyListeners(    )
- *getIgnoreRepaint*  
public boolean getIgnoreRepaint(    )
- *getInputContext*  
public InputContext getInputContext(    )
- *getInputMethodListeners*  
public synchronized InputMethodListener getInputMethodListeners(    )
- *getInputMethodRequests*  
public InputMethodRequests getInputMethodRequests(    )
- *getKeyListeners*  
public synchronized KeyListener getKeyListeners(    )
- *getListeners*  
public EventListener getListeners( java.lang.Class    )
- *getLocale*  
public Locale getLocale(    )
- *getLocation*  
public Point getLocation(    )
- *getLocation*  
public Point getLocation( java.awt.Point    )
- *getLocationOnScreen\_NoTreeLock*  
final Point getLocationOnScreen\_NoTreeLock(    )
- *getLocationOnScreen*  
public Point getLocationOnScreen(    )
- *getMaximumSize*  
public Dimension getMaximumSize(    )
- *getMinimumSize*  
public Dimension getMinimumSize(    )
- *getMouseListeners*  
public synchronized MouseListener getMouseListeners(    )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners(    )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners(    )
- *getName*  
public String getName(    )
- *getNativeContainer*  
Container getNativeContainer(    )

- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )
- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )

- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )



- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )
- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )

- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )

- *removeMouseListener*  
public synchronized void removeMouseListener(  
java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
protected boolean requestFocus( boolean    )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean    , boolean    )
- *requestFocusInWindow*  
public boolean requestFocusInWindow(    )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean    )
- *resetGC*  
void resetGC(    )
- *reshape*  
public void reshape( int    , int    , int    , int    )
- *resize*  
public void resize( java.awt.Dimension    )
- *resize*  
public void resize( int    , int    )
- *setBackground*  
public void setBackground( java.awt.Color    )
- *setBounds*  
public void setBounds( int    , int    , int    , int    )
- *setBounds*  
public void setBounds( java.awt.Rectangle    )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation    )
- *setCursor*  
public void setCursor( java.awt.Cursor    )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget    )
- *setEnabled*  
public void setEnabled( boolean    )

- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusUpCycle*  
public void transferFocusUpCycle( )
- *update*  
public void update( java.awt.Graphics )
- *updateCursorImmediately*  
final void updateCursorImmediately( )
- *validate*  
public void validate( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 6.1.3 CLASS CodePane

---

This is the component that shows and highlights the program while Jeliot is animating, called also code view.

#### DECLARATION

---

```
public class CodePane
extends javax.swing.JComponent
```

#### SERIALIZABLE FIELDS

---

- private LineNumbers nb
  - Line numbering component that handles the correct line numbering in the code view.
- private Font font
  - Font for the code view area.
- private Insets insets
  - Insets for the text. Used for the layout.
- private JScrollPane jsp
  - Pane that handles the scrolling of the code view or the TextArea.

#### FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for gui package.
- private LineNumbers nb
  - Line numbering component that handles the correct line numbering in the code view.
- private Font font
  - Font for the code view area.
- private Insets insets
  - Insets for the text. Used for the layout.
- private JScrollPane jsp
  - Pane that handles the scrolling of the code view or the TextArea.

## CONSTRUCTORS

---

- *CodePane*

**public CodePane( )**

- **Usage**

- \* Constructs the CodePane -object, sets the layout and adds the JScrollPane with JTextArea in the layout.

## METHODS

---

- *calculateLines*

**public int calculateLines( java.lang.String text )**

- **Usage**

- \* Counts how many lines the current program code is taking.

- **Parameters**

- \* **text** - the program code

- **Returns** - the number of lines the given program code takes.

---

- *highlightStatement*

**public void highlightStatement( jeliot.theater.Highlight h )**

- **Usage**

- \* Method highlights the specified Statement area by selecting it.

- **Parameters**

- \* **h** - contains the area that should be highlighted.
- 

- *installProgram*

**public void installProgram( java.lang.String text )**

- **Usage**

- \* Sets the given program code String text into the JTextArea area.

- **Parameters**

- \* **text** - The program code to be set in the JTextArea area.
- 

- *makeScrollPane*

**public JComponent makeScrollPane( )**

- **Usage**

- \* Creates the ScrollPane that shows the line numbering on the left side and the text area in the center.

- **Returns** - the set up scrollpane.

---

- *validateScrollPane*

**public void validateScrollPane( )**

- **Usage**

- \* Validates the scroll pane by setting the correct number of lines to the LineNumbers component.

## METHODS INHERITED FROM CLASS javax.swing.JComponent

- 
- *\_paintImmediately*  
void **\_paintImmediately**( int , int , int , int )
  - *<clinit>*  
static void **<clinit>**( )
  - *addAncestorListener*  
public void **addAncestorListener**( javax.swing.event.AncestorListener )
  - *addNotify*  
public void **addNotify**( )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.lang.String ,  
java.beans.PropertyChangeListener )
  - *addVetoableChangeListener*  
public synchronized void **addVetoableChangeListener**(  
java.beans.VetoableChangeListener )
  - *adjustPaintFlags*  
private void **adjustPaintFlags**( )
  - *alwaysOnTop*  
boolean **alwaysOnTop**( )
  - *checkIfChildObscuredBySibling*  
boolean **checkIfChildObscuredBySibling**( )
  - *componentInputMapChanged*  
void **componentInputMapChanged**( javax.swing.ComponentInputMap )
  - *computeVisibleRect*  
static final void **computeVisibleRect**( java.awt.Component , java.awt.Rectangle )
  - *computeVisibleRect*  
public void **computeVisibleRect**( java.awt.Rectangle )
  - *compWriteObjectNotify*  
void **compWriteObjectNotify**( )
  - *contains*  
public boolean **contains**( int , int )
  - *createToolTip*  
public JToolTip **createToolTip**( )
  - *deregisterNextFocusableComponent*  
private void **deregisterNextFocusableComponent**( )
  - *disable*  
public void **disable**( )
  - *enable*  
public void **enable**( )
  - *enableSerialization*  
void **enableSerialization**( )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , boolean , boolean )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , byte , byte )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , char , char )

- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getAncestorListeners*  
public AncestorListener getAncestorListeners( )
- *getAutoscrolls*  
public boolean getAutoscrolls( )
- *getBorder*  
public Border getBorder( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary getClientProperties( )
- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )
- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )



- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )

- *getToolTipText*  
public String getToolTipText( )
- *getToolTipText*  
public String getToolTipText( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container getTopLevelAncestor( )
- *getTransferHandler*  
public TransferHandler getTransferHandler( )
- *getUIClassID*  
public String getUIClassID( )
- *getVerifyInputWhenFocusTarget*  
public boolean getVerifyInputWhenFocusTarget( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener getVetoableChangeListeners( )
- *getWidth*  
public int getWidth( )
- *getVisibleRect*  
public Rectangle getVisibleRect( )
- *getWriteObjCounter*  
static byte getWriteObjCounter( javax.swing.JComponent )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *grabFocus*  
public void grabFocus( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component )
- *isManagingFocus*  
public boolean isManagingFocus( )
- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )

- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )

- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener(  
java.beans.VetoableChangeListener )
- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )

- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )

- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )

- *adjustDecendantsOnParent*  
void **adjustDecendantsOnParent**( int    )
- *adjustDescendants*  
void **adjustDescendants**( int    )
- *adjustListeningChildren*  
void **adjustListeningChildren**( long    , int    )
- *applyComponentOrientation*  
public void **applyComponentOrientation**( java.awt.ComponentOrientation    )
- *areFocusTraversalKeysSet*  
public boolean **areFocusTraversalKeysSet**( int    )
- *checkGD*  
void **checkGD**( java.lang.String    )
- *clearCurrentFocusCycleRootOnHide*  
void **clearCurrentFocusCycleRootOnHide**(    )
- *clearMostRecentFocusOwnerOnHide*  
void **clearMostRecentFocusOwnerOnHide**(    )
- *containsFocus*  
final boolean **containsFocus**(    )
- *countComponents*  
public int **countComponents**(    )
- *countHierarchyMembers*  
int **countHierarchyMembers**(    )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int    , long    , boolean    )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int    , java.awt.Component    , java.awt.Container    , long    , boolean    )
- *deliverEvent*  
public void **deliverEvent**( java.awt.Event    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchEventToSelf*  
void **dispatchEventToSelf**( java.awt.AWTEvent    )
- *doLayout*  
public void **doLayout**(    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *findComponentAt*  
public Component **findComponentAt**( int    , int    )
- *findComponentAt*  
final Component **findComponentAt**( int    , int    , boolean    )
- *findComponentAt*  
public Component **findComponentAt**( java.awt.Point    )
- *findTraversalRoot*  
private Container **findTraversalRoot**(    )
- *getAccessibleAt*  
Accessible **getAccessibleAt**( java.awt.Point    )
- *getAccessibleChild*  
Accessible **getAccessibleChild**( int    )
- *getAccessibleChildrenCount*  
int **getAccessibleChildrenCount**(    )

- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getComponent*  
public Component **getComponent**( int )
- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentCount*  
public int **getComponentCount**( )
- *getComponents\_NoClientCode*  
final Component **getComponents\_NoClientCode**( )
- *getComponents*  
public Component **getComponents**( )
- *getContainerListeners*  
public synchronized ContainerListener **getContainerListeners**( )
- *getDropTargetEventTarget*  
Component **getDropTargetEventTarget**( int , int , boolean )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy **getFocusTraversalPolicy**( )
- *getInsets*  
public Insets **getInsets**( )
- *getLayout*  
public LayoutManager **getLayout**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )



- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )

- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )
- *removeAll*  
public void **removeAll**(    )
- *removeContainerListener*  
public synchronized void **removeContainerListener**( java.awt.event.ContainerListener    )
- *removeNotify*  
public void **removeNotify**(    )
- *setFocusCycleRoot*  
public void **setFocusCycleRoot**( boolean    )
- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int    , java.util.Set    )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy    )
- *setFont*  
public void **setFont**( java.awt.Font    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setZOrder*  
void **setZOrder**( java.awt.Component    , int    )
- *transferFocusBackward*  
public void **transferFocusBackward**(    )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**(    )
- *update*  
public void **update**( java.awt.Graphics    )
- *validate*  
public void **validate**(    )
- *validateTree*  
protected void **validateTree**(    )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )

METHODS INHERITED FROM CLASS `java.awt.Component`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *action*  
`public boolean action( java.awt.Event , java.lang.Object )`
  - *add*  
`public synchronized void add( java.awt.PopupMenu )`
  - *addComponentListener*  
`public synchronized void addComponentListener( java.awt.event.ComponentListener )`
  - *addFocusListener*  
`public synchronized void addFocusListener( java.awt.event.FocusListener )`
  - *addHierarchyBoundsListener*  
`public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )`
  - *addHierarchyListener*  
`public void addHierarchyListener( java.awt.event.HierarchyListener )`
  - *addInputMethodListener*  
`public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )`
  - *addKeyListener*  
`public synchronized void addKeyListener( java.awt.event.KeyListener )`
  - *addMouseListener*  
`public synchronized void addMouseListener( java.awt.event.MouseListener )`
  - *addMouseMotionListener*  
`public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )`
  - *addMouseWheelListener*  
`public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )`
  - *addNotify*  
`public void addNotify( )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
  - *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
  - *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
  - *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
  - *areInputMethodsEnabled*  
`boolean areInputMethodsEnabled( )`
  - *autoProcessMouseWheel*  
`void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )`
  - *autoTransferFocus*  
`final void autoTransferFocus( boolean )`

- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )

- *dispatchEvent*  
public final void **dispatchEvent**( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean **dispatchMouseWheelToAncestor**( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void **doAutoTransfer**( boolean    )
- *doLayout*  
public void **doLayout**( )
- *enable*  
public void **enable**( )
- *enable*  
public void **enable**( boolean    )
- *enableEvents*  
protected final void **enableEvents**( long    )
- *enableInputMethods*  
public void **enableInputMethods**( boolean    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean **eventTypeEnabled**( int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , int    , int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , java.lang.Object    , java.lang.Object    )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getAccessibleIndexInParent*  
int **getAccessibleIndexInParent**( )
- *getAccessibleStateSet*  
AccessibleStateSet **getAccessibleStateSet**( )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle    )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )

- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )

- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener **getMouseMotionListeners**( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener **getMouseWheelListeners**( )
- *getName*  
public String **getName**( )
- *getNativeContainer*  
Container **getNativeContainer**( )
- *getParent\_NoClientCode*  
final Container **getParent\_NoClientCode**( )
- *getParent*  
public Container **getParent**( )
- *getPeer*  
public ComponentPeer **getPeer**( )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getSize*  
public Dimension **getSize**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getToolkitImpl*  
final Toolkit **getToolkitImpl**( )
- *getTreeLock*  
public final Object **getTreeLock**( )
- *getWidth*  
public int **getWidth**( )

- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *gotFocus*  
public boolean **gotFocus**( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean **handleEvent**( java.awt.Event )
- *hasFocus*  
public boolean **hasFocus**( )
- *hide*  
public void **hide**( )
- *imageUpdate*  
public boolean **imageUpdate**( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *inside*  
public boolean **inside**( int , int )
- *invalidate*  
public void **invalidate**( )
- *isBackgroundSet*  
public boolean **isBackgroundSet**( )
- *isCursorSet*  
public boolean **isCursorSet**( )
- *isDisplayable*  
public boolean **isDisplayable**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isEnabled*  
public boolean **isEnabled**( )
- *isEnabledImpl*  
final boolean **isEnabledImpl**( )
- *isFocusable*  
public boolean **isFocusable**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusOwner*  
public boolean **isFocusOwner**( )
- *isFocusTraversable*  
public boolean **isFocusTraversable**( )
- *isFocusTraversableOverridden*  
final boolean **isFocusTraversableOverridden**( )
- *isFontSet*  
public boolean **isFontSet**( )
- *isForegroundSet*  
public boolean **isForegroundSet**( )
- *isLightweight*  
public boolean **isLightweight**( )



- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )

- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvent*  
boolean postsOldMouseEvent( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )

- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *remove*  
public synchronized void remove( java.awt.MenuComponent    )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener    )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener    )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener    )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener    )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener    )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener    )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener    )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener    )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    , java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
protected boolean requestFocus( boolean    )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean    , boolean    )

- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )

- *setSize*  
public void setSize( java.awt.Dimension    )
- *setSize*  
public void setSize( int    , int    )
- *setVisible*  
public void setVisible( boolean    )
- *show*  
public void show(    )
- *show*  
public void show( boolean    )
- *size*  
public Dimension size(    )
- *toString*  
public String toString(    )
- *transferFocus*  
public void transferFocus(    )
- *transferFocusBackward*  
public void transferFocusBackward(    )
- *transferFocusUpCycle*  
public void transferFocusUpCycle(    )
- *update*  
public void update( java.awt.Graphics    )
- *updateCursorImmediately*  
final void updateCursorImmediately(    )
- *validate*  
public void validate(    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

#### 6.1.4 CLASS DraggableComponent

---

Not currently used in Jeliot 3. The design of the draggable component that perhaps will be added inside Jeliot's next version.

##### DECLARATION

---

```
public class DraggableComponent
extends javax.swing.JPanel
implements java.awt.event.MouseListener, java.awt.event.MouseMotionListener
```

##### CONSTRUCTORS

---

- *DraggableComponent*  
protected **DraggableComponent**(    )

##### – Usage

- \* Constructs the DraggableComponent -objects. Adds MouseListener and MouseMotionListener. Sets new cursor for the component area.

METHODS

---

• *inStretchArea*

**boolean inStretchArea( java.awt.Point p )**

– **Usage**

\* Checks if the given Point p is inside that area where you resize the component.

– **Parameters**

\* p - The point consist of the coordinates of the cursor when the mouse was clicked.

– **Returns** - True if the cursor is on the resize area and false when it isn't on that area.

---

• *isFree*

**public boolean isFree( )**

– **Usage**

\* Returns the value wheter the mode -variable is FREE (True) or something else.  
(False)

---

• *mouseClicked*

**public void mouseClicked( java.awt.event.MouseEvent e )**

---

• *mouseDragged*

**public void mouseDragged( java.awt.event.MouseEvent e )**

– **Usage**

\* Handles the mouse event when the mouse is dragged. If the mode is DRAG then this method moves the component as dragged. If the mode is RESIZE then this method resizes the component as dragged.

– **Parameters**

\* e - Mouse event of the mouse dragging.

---

• *mouseEntered*

**public void mouseEntered( java.awt.event.MouseEvent e )**

---

• *mouseExited*

**public void mouseExited( java.awt.event.MouseEvent e )**

---

• *mouseMoved*

**public void mouseMoved( java.awt.event.MouseEvent e )**

---

• *mousePressed*

**public void mousePressed( java.awt.event.MouseEvent e )**

– **Usage**

\* Handles the event when the mouse's left button is pressed. Tests if user wants to resize or drag the component and changes the mode in that way.

– **Parameters**

\* e - Mouse event of the mouse button pressing.

---

• *mouseReleased*

**public void mouseReleased( java.awt.event.MouseEvent e )**

– **Usage**

\* Changes the mode back to FREE and revalidates the screen.

– **Parameters**

\* **e** - Mouse event of the mouse released.

METHODS INHERITED FROM CLASS javax.swing.JPanel

---

- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getUI*  
public PanelUI getUI( )
- *getUIClassID*  
public String getUIClassID( )
- *paramString*  
protected String paramString( )
- *setUI*  
public void setUI( javax.swing.plaf.PanelUI )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

METHODS INHERITED FROM CLASS javax.swing.JComponent

---

- *\_paintImmediately*  
void \_paintImmediately( int , int , int , int )
- *<clinit>*  
static void <clinit>( )
- *addAncestorListener*  
public void addAncestorListener( javax.swing.event.AncestorListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *addVetoableChangeListener*  
public synchronized void addVetoableChangeListener( java.beans.VetoableChangeListener )
- *adjustPaintFlags*  
private void adjustPaintFlags( )
- *alwaysOnTop*  
boolean alwaysOnTop( )
- *checkIfChildObscuredBySibling*  
boolean checkIfChildObscuredBySibling( )
- *componentInputMapChanged*  
void componentInputMapChanged( javax.swing.ComponentInputMap )

- *computeVisibleRect*  
static final void computeVisibleRect( java.awt.Component , java.awt.Rectangle )
- *computeVisibleRect*  
public void computeVisibleRect( java.awt.Rectangle )
- *compWriteObjectNotify*  
void compWriteObjectNotify( )
- *contains*  
public boolean contains( int , int )
- *createToolTip*  
public JToolTip createToolTip( )
- *deregisterNextFocusableComponent*  
private void deregisterNextFocusableComponent( )
- *disable*  
public void disable( )
- *enable*  
public void enable( )
- *enableSerialization*  
void enableSerialization( )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , byte , byte )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , char , char )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )



- *getAlignmentY*  
public float **getAlignmentY**( )
- *getAncestorListeners*  
public AncestorListener **getAncestorListeners**( )
- *getAutoscrolls*  
public boolean **getAutoscrolls**( )
- *getBorder*  
public Border **getBorder**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary **getClientProperties**( )
- *getClientProperty*  
public final Object **getClientProperty**( java.lang.Object )
- *getComponentGraphics*  
protected Graphics **getComponentGraphics**( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int **getConditionForKeyStroke**( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean **getCreatedDoubleBuffer**( boolean )
- *getDebugGraphicsOptions*  
public int **getDebugGraphicsOptions**( )
- *getDefaultLocale*  
public static Locale **getDefaultLocale**( )
- *getFlag*  
private boolean **getFlag**( int )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getHeight*  
public int **getHeight**( )
- *getInputMap*  
public final InputMap **getInputMap**( )
- *getInputMap*  
public final InputMap **getInputMap**( int )
- *getInputMap*  
final InputMap **getInputMap**( int , boolean )
- *getInputVerifier*  
public InputVerifier **getInputVerifier**( )
- *getInsets*  
public Insets **getInsets**( )
- *getInsets*  
public Insets **getInsets**( java.awt.Insets )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set **getManagingFocusBackwardTraversalKeys**( )
- *getManagingFocusForwardTraversalKeys*  
static Set **getManagingFocusForwardTraversalKeys**( )

- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getNextFocusableComponent*  
public Component **getNextFocusableComponent**( )
- *getObscuredState*  
private int **getObscuredState**( int , int , int , int , int )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke **getRegisteredKeyStrokes**( )
- *getRootPane*  
public JRootPane **getRootPane**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean **getSuppressDropTarget**( )
- *getToolTipLocation*  
public Point **getToolTipLocation**( java.awt.event.MouseEvent )
- *getToolTipText*  
public String **getToolTipText**( )
- *getToolTipText*  
public String **getToolTipText**( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container **getTopLevelAncestor**( )
- *getTransferHandler*  
public TransferHandler **getTransferHandler**( )
- *getUIClassID*  
public String **getUIClassID**( )
- *getVerifyInputWhenFocusTarget*  
public boolean **getVerifyInputWhenFocusTarget**( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener **getVetoableChangeListeners**( )
- *getWidth*  
public int **getWidth**( )
- *getVisibleRect*  
public Rectangle **getVisibleRect**( )
- *getWriteObjCounter*  
static byte **getWriteObjCounter**( javax.swing.JComponent )
- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *grabFocus*  
public void **grabFocus**( )

- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component    )
- *isManagingFocus*  
public boolean isManagingFocus( )
- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )
- *paint*  
public void paint( java.awt.Graphics    )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics    )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics    )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics    )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent    , java.awt.Component    , java.awt.Graphics    , int    , int    , int    , int    )
- *paintImmediately*  
public void paintImmediately( int    , int    , int    , int    )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle    )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent    , java.awt.Graphics    , int    , int    , int    , int    , java.awt.Image    )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics    )
- *printAll*  
public void printAll( java.awt.Graphics    )
- *printBorder*  
protected void printBorder( java.awt.Graphics    )

- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener , javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener , java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener( java.beans.VetoableChangeListener )

- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )
- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )

- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )
- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

## METHODS INHERITED FROM CLASS java.awt.Container

- 
- *<clinit>*  
static void <clinit>( )
  - *add*  
public Component add( java.awt.Component )
  - *add*  
public Component add( java.awt.Component , int )
  - *add*  
public void add( java.awt.Component , java.lang.Object )
  - *add*  
public void add( java.awt.Component , java.lang.Object , int )
  - *add*  
public Component add( java.lang.String , java.awt.Component )
  - *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
  - *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
  - *addNotify*  
public void addNotify( )
  - *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
  - *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
  - *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int )
  - *adjustDescendants*  
void adjustDescendants( int )
  - *adjustListeningChildren*  
void adjustListeningChildren( long , int )
  - *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
  - *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
  - *checkGD*  
void checkGD( java.lang.String )
  - *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
  - *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
  - *containsFocus*  
final boolean containsFocus( )
  - *countComponents*  
public int countComponents( )
  - *countHierarchyMembers*  
int countHierarchyMembers( )
  - *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )

- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )



- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseEventTarget*  
Component getMouseEventTarget( int , int , boolean )
- *getMouseEventTarget*  
private Component getMouseEventTarget( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component getMouseEventTargetImpl( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *insets*  
public Insets insets( )
- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )

- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )
- *printComponents*  
public void printComponents( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processContainerEvent*  
protected void processContainerEvent( java.awt.event.ContainerEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *proxyEnableEvents*  
void proxyEnableEvents( long )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public void remove( java.awt.Component )
- *remove*  
public void remove( int )
- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )

- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void setFocusTraversalPolicy( java.awt.FocusTraversalPolicy )
- *setFont*  
public void setFont( java.awt.Font )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setZOrder*  
void setZOrder( java.awt.Component , int )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusDownCycle*  
public void transferFocusDownCycle( )
- *update*  
public void update( java.awt.Graphics )
- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )
- *addInputMethodListener*  
public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )

- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *areInputMethodsEnabled*  
boolean areInputMethodsEnabled( )
- *autoProcessMouseWheel*  
void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )
- *autoTransferFocus*  
final void autoTransferFocus( boolean )
- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )

- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *doAutoTransfer*  
private void doAutoTransfer( boolean )
- *doLayout*  
public void doLayout( )
- *enable*  
public void enable( )
- *enable*  
public void enable( boolean )
- *enableEvents*  
protected final void enableEvents( long )
- *enableInputMethods*  
public void enableInputMethods( boolean )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , int , int )

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getAccessibleIndexInParent*  
int **getAccessibleIndexInParent**( )
- *getAccessibleStateSet*  
AccessibleStateSet **getAccessibleStateSet**( )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )
- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )

- *getFontMetrics*  
public FontMetrics getFontMetrics( java.awt.Font    )
- *getForeground*  
public Color getForeground( )
- *getGraphics*  
public Graphics getGraphics( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getHeight*  
public int getHeight( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener getHierarchyBoundsListeners( )
- *getHierarchyListeners*  
public synchronized HierarchyListener getHierarchyListeners( )
- *getIgnoreRepaint*  
public boolean getIgnoreRepaint( )
- *getInputContext*  
public InputContext getInputContext( )
- *getInputMethodListeners*  
public synchronized InputMethodListener getInputMethodListeners( )
- *getInputMethodRequests*  
public InputMethodRequests getInputMethodRequests( )
- *getKeyListeners*  
public synchronized KeyListener getKeyListeners( )
- *getListeners*  
public EventListener getListeners( java.lang.Class    )
- *getLocale*  
public Locale getLocale( )
- *getLocation*  
public Point getLocation( )
- *getLocation*  
public Point getLocation( java.awt.Point    )
- *getLocationOnScreen\_NoTreeLock*  
final Point getLocationOnScreen\_NoTreeLock( )
- *getLocationOnScreen*  
public Point getLocationOnScreen( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseListeners*  
public synchronized MouseListener getMouseListeners( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )

- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )
- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )



- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )

- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )
- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )

- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )

- *removeMouseListener*  
public synchronized void removeMouseListener(  
java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
protected boolean requestFocus( boolean    )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean    , boolean    )
- *requestFocusInWindow*  
public boolean requestFocusInWindow(    )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean    )
- *resetGC*  
void resetGC(    )
- *reshape*  
public void reshape( int    , int    , int    , int    )
- *resize*  
public void resize( java.awt.Dimension    )
- *resize*  
public void resize( int    , int    )
- *setBackground*  
public void setBackground( java.awt.Color    )
- *setBounds*  
public void setBounds( int    , int    , int    , int    )
- *setBounds*  
public void setBounds( java.awt.Rectangle    )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation    )
- *setCursor*  
public void setCursor( java.awt.Cursor    )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget    )
- *setEnabled*  
public void setEnabled( boolean    )

- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusUpCycle*  
public void transferFocusUpCycle( )
- *update*  
public void update( java.awt.Graphics )
- *updateCursorImmediately*  
final void updateCursorImmediately( )
- *validate*  
public void validate( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 6.1.5 CLASS **HelpWindow**

---

The design of the help window.

#### DECLARATION

---

```
public class HelpWindow
extends javax.swing.JFrame
```

#### SERIALIZABLE FIELDS

---

- private `JEditorPane` `editorPane`
  - The pane where helping information will be shown.
- private `JScrollPane` `jsp`
  - The pane that handles the scrolling of the editor pane showing the content.

#### FIELDS

---

- private static `ResourceBundle` `bundle`
  - The resource bundle for gui package.
- private `JEditorPane` `editorPane`
  - The pane where helping information will be shown.
- private `JScrollPane` `jsp`
  - The pane that handles the scrolling of the editor pane showing the content.

#### CONSTRUCTORS

---

- *HelpWindow*  
`public HelpWindow( java.awt.Image icon, java.lang.String udir )`
  - **Usage**
    - \* constructs the `HelpWindow` by creating a `JFrame`. Sets inside the `JFrame` `JScrollPane` with `JEditorPane` `editorPane`. Sets the size of the `JFrame` as 400 x 600
  - **Parameters**
    - \* `icon` - Icon to be shown in the upper right corner of the window.
    - \* `udir` - directory of the current invocation

METHODS

---

- *showURL*  
`public void showURL( java.net.URL url )`
  - **Usage**
    - \* Shows the given url in the editor pane.
  - **Parameters**
    - \* `url` - The document in the url will be showed in JEditorPane editorPane.

METHODS INHERITED FROM CLASS `javax.swing.JFrame`

---

- *<clinit>*  
`static void <clinit>( )`
- *addImpl*  
`protected void addImpl( java.awt.Component , java.lang.Object , int )`
- *createRootPane*  
`protected JRootPane createRootPane( )`
- *createRootPaneException*  
`private Error createRootPaneException( java.lang.String )`
- *frameInit*  
`protected void frameInit( )`
- *getAccessibleContext*  
`public AccessibleContext getAccessibleContext( )`
- *getContentPane*  
`public Container getContentPane( )`
- *getDefaultCloseOperation*  
`public int getDefaultCloseOperation( )`
- *getGlassPane*  
`public Component getGlassPane( )`
- *getJMenuBar*  
`public JMenuBar getJMenuBar( )`
- *getLayeredPane*  
`public JLayeredPane getLayeredPane( )`
- *getRootPane*  
`public JRootPane getRootPane( )`
- *isDefaultLookAndFeelDecorated*  
`public static boolean isDefaultLookAndFeelDecorated( )`
- *isRootPaneCheckingEnabled*  
`protected boolean isRootPaneCheckingEnabled( )`
- *paramString*  
`protected String paramString( )`
- *processWindowEvent*  
`protected void processWindowEvent( java.awt.event.WindowEvent )`
- *remove*  
`public void remove( java.awt.Component )`
- *setContentPane*  
`public void setContentPane( java.awt.Container )`

- *setDefaultCloseOperation*  
public void setDefaultCloseOperation( int )
- *setDefaultLookAndFeelDecorated*  
public static void setDefaultLookAndFeelDecorated( boolean )
- *setGlassPane*  
public void setGlassPane( java.awt.Component )
- *setJMenuBar*  
public void setJMenuBar( javax.swing.JMenuBar )
- *setLayeredPane*  
public void setLayeredPane( javax.swing.JLayeredPane )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setRootPane*  
protected void setRootPane( javax.swing.JRootPane )
- *setRootPaneCheckingEnabled*  
protected void setRootPaneCheckingEnabled( boolean )
- *update*  
public void update( java.awt.Graphics )

#### METHODS INHERITED FROM CLASS java.awt.Frame

---

- *<clinit>*  
static void <clinit>( )
- *addNotify*  
public void addNotify( )
- *addToFrameList*  
void addToFrameList( )
- *constructComponentName*  
String constructComponentName( )
- *finalize*  
protected void finalize( )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getCursorType*  
public int getCursorType( )
- *getExtendedState*  
public synchronized int getExtendedState( )
- *getFrames*  
public static Frame getFrames( )
- *getIconImage*  
public Image getIconImage( )
- *getMaximizedBounds*  
public Rectangle getMaximizedBounds( )
- *getMenuBar*  
public MenuBar getMenuBar( )
- *getState*  
public synchronized int getState( )
- *getTitle*  
public String getTitle( )



- *init*  
private void **init**( java.lang.String , java.awt.GraphicsConfiguration )
- *initIDs*  
private static native void **initIDs**( )
- *isResizable*  
public boolean **isResizable**( )
- *isUndecorated*  
public boolean **isUndecorated**( )
- *paramString*  
protected String **paramString**( )
- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *remove*  
public void **remove**( java.awt.MenuComponent )
- *removeFromFrameList*  
void **removeFromFrameList**( )
- *removeNotify*  
public void **removeNotify**( )
- *setCursor*  
public void **setCursor**( int )
- *setExtendedState*  
public synchronized void **setExtendedState**( int )
- *setIconImage*  
public synchronized void **setIconImage**( java.awt.Image )
- *setMaximizedBounds*  
public synchronized void **setMaximizedBounds**( java.awt.Rectangle )
- *setMenuBar*  
public void **setMenuBar**( java.awt.MenuBar )
- *setResizable*  
public void **setResizable**( boolean )
- *setState*  
public synchronized void **setState**( int )
- *setTitle*  
public void **setTitle**( java.lang.String )
- *setUndecorated*  
public void **setUndecorated**( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

METHODS INHERITED FROM CLASS `java.awt.Window`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *addNotify*  
`public void addNotify( )`
  - *addOwnedWindow*  
`void addOwnedWindow( java.lang.ref.WeakReference )`
  - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )`
  - *addWindowFocusListener*  
`public synchronized void addWindowFocusListener(  
java.awt.event.WindowFocusListener )`
  - *addWindowListener*  
`public synchronized void addWindowListener( java.awt.event.WindowListener )`
  - *addWindowStateListener*  
`public synchronized void addWindowStateListener(  
java.awt.event.WindowStateListener )`
  - *adjustDecendantsOnParent*  
`void adjustDecendantsOnParent( int )`
  - *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
  - *applyResourceBundle*  
`public void applyResourceBundle( java.util.ResourceBundle )`
  - *applyResourceBundle*  
`public void applyResourceBundle( java.lang.String )`
  - *clearMostRecentFocusOwnerOnHide*  
`final void clearMostRecentFocusOwnerOnHide( )`
  - *connectOwnedWindow*  
`void connectOwnedWindow( java.awt.Window )`
  - *constructComponentName*  
`String constructComponentName( )`
  - *createBufferStrategy*  
`public void createBufferStrategy( int )`
  - *createBufferStrategy*  
`public void createBufferStrategy( int , java.awt.BufferCapabilities )`
  - *deliverMouseWheelToAncestor*  
`void deliverMouseWheelToAncestor( java.awt.event.MouseWheelEvent )`
  - *dispatchEventImpl*  
`void dispatchEventImpl( java.awt.AWTEvent )`
  - *dispatchMouseWheelToAncestor*  
`boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )`
  - *dispose*  
`public void dispose( )`
  - *eventEnabled*  
`boolean eventEnabled( java.awt.AWTEvent )`
  - *finalize*  
`protected void finalize( )`

- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getBufferStrategy*  
public BufferStrategy getBufferStrategy( )
- *getFocusableWindowState*  
public boolean getFocusableWindowState( )
- *getFocusCycleRootAncestor*  
public final Container getFocusCycleRootAncestor( )
- *getFocusOwner*  
public Component getFocusOwner( )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getInputContext*  
public InputContext getInputContext( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocale*  
public Locale getLocale( )
- *getMostRecentFocusOwner*  
public Component getMostRecentFocusOwner( )
- *getOwnedWindows*  
public Window getOwnedWindows( )
- *getOwner*  
public Window getOwner( )
- *getTemporaryLostComponent*  
Component getTemporaryLostComponent( )
- *getToolkit*  
public Toolkit getToolkit( )
- *getWarningString*  
public final String getWarningString( )
- *getWindowFocusListeners*  
public synchronized WindowFocusListener getWindowFocusListeners( )
- *getWindowListeners*  
public synchronized WindowListener getWindowListeners( )
- *getWindowStateListeners*  
public synchronized WindowStateListener getWindowStateListeners( )
- *hide*  
public void hide( )
- *init*  
private void init( java.awt.GraphicsConfiguration )
- *initIDs*  
private static native void initIDs( )
- *isActive*  
public boolean isActive( )
- *isFocusableWindow*  
public final boolean isFocusableWindow( )
- *isFocusCycleRoot*  
public final boolean isFocusCycleRoot( )

- *isFocused*  
public boolean isFocused( )
- *isShowing*  
public boolean isShowing( )
- *ownedInit*  
private void ownedInit( java.awt.Window )
- *pack*  
public void pack( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postWindowEvent*  
synchronized void postWindowEvent( int )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processWindowEvent*  
protected void processWindowEvent( java.awt.event.WindowEvent )
- *processWindowFocusEvent*  
protected void processWindowFocusEvent( java.awt.event.WindowEvent )
- *processWindowStateEvent*  
protected void processWindowStateEvent( java.awt.event.WindowEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removeOwnedWindow*  
void removeOwnedWindow( java.lang.ref.WeakReference )
- *removeWindowFocusListener*  
public synchronized void removeWindowFocusListener( java.awt.event.WindowFocusListener )
- *removeWindowListener*  
public synchronized void removeWindowListener( java.awt.event.WindowListener )
- *removeWindowStateListener*  
public synchronized void removeWindowStateListener( java.awt.event.WindowStateListener )
- *resetGC*  
void resetGC( )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setFocusableWindowState*  
public void setFocusableWindowState( boolean )
- *setFocusCycleRoot*  
public final void setFocusCycleRoot( boolean )
- *setLocationRelativeTo*  
public void setLocationRelativeTo( java.awt.Component )
- *setTemporaryLostComponent*  
Component setTemporaryLostComponent( java.awt.Component )
- *setWarningString*  
private void setWarningString( )

- *show*  
public void show( )
- *toBack*  
public void toBack( )
- *toFront*  
public void toFront( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int )
- *adjustDescendants*  
void adjustDescendants( int )
- *adjustListeningChildren*  
void adjustListeningChildren( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *checkGD*  
void checkGD( java.lang.String )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )

- *containsFocus*  
final boolean containsFocus( )
- *countComponents*  
public int countComponents( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )

- *getContainerListeners*  
public synchronized ContainerListener **getContainerListeners**( )
- *getDropTargetEventTarget*  
Component **getDropTargetEventTarget**( int , int , boolean )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy **getFocusTraversalPolicy**( )
- *getInsets*  
public Insets **getInsets**( )
- *getLayout*  
public LayoutManager **getLayout**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )
- *invalidate*  
public void **invalidate**( )
- *invalidateTree*  
void **invalidateTree**( )
- *isAncestorOf*  
public boolean **isAncestorOf**( java.awt.Component )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean **isFocusTraversalPolicySet**( )
- *isParentOf*  
boolean **isParentOf**( java.awt.Component )
- *layout*  
public void **layout**( )

- *lightweightPaint*  
void **lightweightPaint**( java.awt.Graphics    )
- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics    )
- *list*  
public void **list**( java.io.PrintStream    , int    )
- *list*  
public void **list**( java.io.PrintWriter    , int    )
- *locate*  
public Component **locate**( int    , int    )
- *minimumSize*  
public Dimension **minimumSize**( )
- *nextFocusHelper*  
boolean **nextFocusHelper**( )
- *numListening*  
int **numListening**( long    )
- *paint*  
public void **paint**( java.awt.Graphics    )
- *paintComponents*  
public void **paintComponents**( java.awt.Graphics    )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics    )
- *paramString*  
protected String **paramString**( )
- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent    )
- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *preProcessKeyEvent*  
void **preProcessKeyEvent**( java.awt.event.KeyEvent    )
- *print*  
public void **print**( java.awt.Graphics    )
- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )



- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void setFocusTraversalPolicy( java.awt.FocusTraversalPolicy )
- *setFont*  
public void setFont( java.awt.Font )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setZOrder*  
void setZOrder( java.awt.Component , int )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusDownCycle*  
public void transferFocusDownCycle( )
- *update*  
public void update( java.awt.Graphics )
- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )

- *addInputMethodListener*  
public synchronized void addInputMethodListener(  
java.awt.event.InputMethodListener    )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener    )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener    )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener(  
java.awt.event.MouseMotionListener    )
- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener  
)
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long    , int    )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation    )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int    )
- *areInputMethodsEnabled*  
boolean areInputMethodsEnabled( )
- *autoProcessMouseWheel*  
void autoProcessMouseWheel( java.awt.event.MouseWheelEvent    )
- *autoTransferFocus*  
final void autoTransferFocus( boolean    )
- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String    )
- *checkImage*  
public int checkImage( java.awt.Image    , java.awt.image.ImageObserver    )
- *checkImage*  
public int checkImage( java.awt.Image    , int    , int    , java.awt.image.ImageObserver  
)
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent    , java.awt.AWTEvent    )
- *constructComponentName*  
String constructComponentName( )

- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *doAutoTransfer*  
private void doAutoTransfer( boolean )
- *doLayout*  
public void doLayout( )
- *enable*  
public void enable( )
- *enable*  
public void enable( boolean )
- *enableEvents*  
protected final void enableEvents( long )
- *enableInputMethods*  
public void enableInputMethods( boolean )

- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent( )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getBackBuffer*  
Image getBackBuffer( )
- *getBackground*  
public Color getBackground( )
- *getBounds*  
public Rectangle getBounds( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy getBufferStrategy( )
- *getColorModel*  
public ColorModel getColorModel( )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener getComponentListeners( )
- *getComponentOrientation*  
public ComponentOrientation getComponentOrientation( )
- *getCursor*  
public Cursor getCursor( )
- *getDropTarget*  
public synchronized DropTarget getDropTarget( )
- *getFocusCycleRootAncestor*  
public Container getFocusCycleRootAncestor( )
- *getFocusListeners*  
public synchronized FocusListener getFocusListeners( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set getFocusTraversalKeys\_NoIDCheck( int )

- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int    )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font    )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class    )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point    )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )

- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )
- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )
- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )

- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )
- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )

- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics    )
- *list*  
public void **list**(    )
- *list*  
public void **list**( java.io.PrintStream    )
- *list*  
public void **list**( java.io.PrintStream    , int    )
- *list*  
public void **list**( java.io.PrintWriter    )
- *list*  
public void **list**( java.io.PrintWriter    , int    )
- *locate*  
public Component **locate**( int    , int    )
- *location*  
public Point **location**(    )
- *lostFocus*  
public boolean **lostFocus**( java.awt.Event    , java.lang.Object    )
- *minimumSize*  
public Dimension **minimumSize**(    )
- *mouseDown*  
public boolean **mouseDown**( java.awt.Event    , int    , int    )
- *mouseDrag*  
public boolean **mouseDrag**( java.awt.Event    , int    , int    )
- *mouseEnter*  
public boolean **mouseEnter**( java.awt.Event    , int    , int    )
- *mouseExit*  
public boolean **mouseExit**( java.awt.Event    , int    , int    )
- *mouseMove*  
public boolean **mouseMove**( java.awt.Event    , int    , int    )
- *mouseUp*  
public boolean **mouseUp**( java.awt.Event    , int    , int    )
- *move*  
public void **move**( int    , int    )
- *nextFocus*  
public void **nextFocus**(    )
- *nextFocusHelper*  
boolean **nextFocusHelper**(    )
- *numListening*  
int **numListening**( long    )
- *paint*  
public void **paint**( java.awt.Graphics    )
- *paintAll*  
public void **paintAll**( java.awt.Graphics    )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics    )
- *paramString*  
protected String **paramString**(    )
- *postEvent*  
public boolean **postEvent**( java.awt.Event    )



- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *prepareImage*  
public boolean **prepareImage**( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean **prepareImage**( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void **print**( java.awt.Graphics )
- *printAll*  
public void **printAll**( java.awt.Graphics )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics )
- *processComponentEvent*  
protected void **processComponentEvent**( java.awt.event.ComponentEvent )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent )
- *processFocusEvent*  
protected void **processFocusEvent**( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void **processHierarchyBoundsEvent**( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void **processHierarchyEvent**( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void **processInputMethodEvent**( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void **processKeyEvent**( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void **processMouseEvent**( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void **processMouseMotionEvent**( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void **processMouseWheelEvent**( java.awt.event.MouseWheelEvent )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *remove*  
public synchronized void **remove**( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void **removeComponentListener**( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void **removeFocusListener**( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void **removeHierarchyBoundsListener**( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void **removeHierarchyListener**( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void **removeInputMethodListener**( java.awt.event.InputMethodListener )

- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *repaint*  
public void repaint( )
- *repaint*  
public void repaint( int , int , int , int )
- *repaint*  
public void repaint( long )
- *repaint*  
public void repaint( long , int , int , int , int )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )

- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )

- *transferFocusUpCycle*  
public void **transferFocusUpCycle**( )
- *update*  
public void **update**( java.awt.Graphics    )
- *updateCursorImmediately*  
final void **updateCursorImmediately**( )
- *validate*  
public void **validate**( )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )

### 6.1.6 CLASS JavaFileFilter

---

Filter for the file chooser to only show the java source code files. Modified from the example for the Sun's Java Website.

#### DECLARATION

---

```
public class JavaFileFilter
extends javax.swing.filechooser.FileFilter
```

#### FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for gui package

#### CONSTRUCTORS

---

- *JavaFileFilter*  
public **JavaFileFilter**( )

#### METHODS

---

- *accept*  
public boolean **accept**( java.io.File f )
  - **Usage**
    - \* Method accepts only the files whose extension which is defined in the resource bundle with name extension.java
  - **See Also**
    - \* javax.swing.filechooser.FileFilter.accept(java.io.File)
- *getDescription*  
public String **getDescription**( )

- **Usage**

- \* Returns the human readable description of this filter described in the resource bundle as extension.java.description.

- **See Also**

- \* `javax.swing.filechooser.FileFilter.getDescription()`

---

- *getExtension*

```
public String getExtension( java.io.File f )
```

- **Usage**

- \* Return the extension portion of the file's name .

- **Parameters**

- \* **f** - The file whose extension is needed.

- **Returns** - the string presentation of the files extension.

METHODS INHERITED FROM CLASS `javax.swing.filechooser.FileFilter`

---

- *accept*

```
public abstract boolean accept( java.io.File f )
```

- *getDescription*

```
public abstract String getDescription( )
```

### 6.1.7 CLASS JeliotWindow

---

The main window of the Jeliot 3.

DECLARATION

---

```
public class JeliotWindow
extends java.lang.Object
```

FIELDS

---

- private static ResourceBundle bundle

- The resource bundle for gui package

- private String jeliotVersion

- The version information about Jeliot from name and version from the resource bundle.

- private boolean showMessagesInDialogs

- Should the messages during the program visualization be shown as message dialogs.

- private boolean errorOccured

- True if an error has occurred during the execution and false if no error was encountered.

- private AboutWindow aw

- The about window of Jeliot 3.
- private HelpWindow hw
  - The help window of Jeliot 3.
- private int previousSpeed
  - The previous speed before the run until line is set.
- private JFrame frame
  - The frame in which all the action goes on.
- private Theater theatre
  - The theatre in which the programs are animated.
- private AnimationEngine engine
  - The animation engine that that will animate the code.
- private Jeliot jeliot
  - The main program.
- private CodePane codePane
  - The code pane where the code is shown during the animation.
- private CodeEditor editor
  - The code editor in which the users can write their code.
- private JSplitPane codeNest
  - The pane that splits the window.
- private JButton stepButton
  - The step button.
- private JButton playButton
  - The play button.
- private JButton pauseButton
  - The pause button.
- private JButton rewindButton
  - The rewind button.
- private JButton editButton
  - The edit button.
- private JButton compileButton
  - The compile button.
- private JSlider speedSlider
  - Slider tha controls the animation speed.
- private JTextArea outputConsole

- In this text area will come the output of the user-made programs.
- private Vector animationMenuItems
  - Menu items that should be either enabled or disabled when the animation mode is entered or exited respectively.
- private ImageLoader iLoad
  - This ImageLoader will load all the images.
- private PanelController panelController
  - This variable will control the panels.
- private JEditorPane errorJEditorPane
  - This JEditorPane errorJEditorPane will show the error messages for the users.
- private JScrollPane errorPane
  - Scroll pane that provides the scroll bars for the error pane’s editor
- private JPanel errorViewer
  - This JPanel errorViewer will help the showing of the error messages for the users.
- private ActionListener stepAction
  - Action listeners for the step- button.
- private ActionListener playAction
  - Action listeners for the play- button.
- private ActionListener pauseAction
  - Action listeners for the pause- button.
- private ActionListener rewindAction
  - Action listeners for the rewind- button.
- private ActionListener exit
  - Action listener for the exit.
- private Vector editWidgets
  - Helping to enable and disable the components.
- private Vector animWidgets
  - Helping to enable and disable the components.
- private String udir
  - The user directory.

---

## CONSTRUCTORS

- *JeliotWindow*

```
public JeliotWindow( jeliot.Jeliot  jeliot, jeliot.gui.CodePane  codePane,
jeliot.theater.Theater  theatre, jeliot.theater.AnimationEngine  engine,
jeliot.theater.ImageLoader  iLoad, java.lang.String  udir )
```

- **Usage**

- \* Assigns the values of the parameters in the object values. Constructs the panelController with theatre and iload.

- **Parameters**

- \* **jeliot** - The main program.
- \* **codePane** - The pane where all the code is shown while animated.
- \* **theatre** - The theatre where all the code is animated.
- \* **engine** - The engine that animates the code.
- \* **iLoad** - The imageloader that loads all the images.
- \* **udir** - The user directory

---

## METHODS

- *addInAnimationMenuItems*

```
public void addInAnimationMenuItems( javax.swing.JMenu [] jm )
```

- **Usage**

- \* Adds the given JMenu's JMenuItem's into the Vector animationMenuItems.

---

- *animationFinished*

```
public void animationFinished( )
```

- **Usage**

- \* Changes the user interface when the animation is finished.

---

- *changeCodePane*

```
private void changeCodePane( javax.swing.JComponent  comp )
```

- **Usage**

- \* Changes the code pane in the codeNest. Sets inside the codeNest the new code pane.

- **Parameters**

- \* **comp** - The component that is changes in the code pane.

---

- *changeTheatrePane*

```
private void changeTheatrePane( javax.swing.JComponent  comp )
```

- **Usage**

- \* Changes the theatre pane in the codeNest. Sets inside the codeNest the new theatre pane.

- **Parameters**

- \* **comp** - The component that is changes in the theatre pane.

---



- *enableWidgets*

```
private void enableWidgets( java.util.Enumeration  enum, boolean  enable )
```

- **Usage**

- \* Enables or disables the components depending on the second parameter.

- **Parameters**

- \* **enum** - The collection of components that are set enabled or disabled as the boolean enable is set.
    - \* **enable** - Sets wheter the components are enabled or disabled.

---

- *enterAnimate*

```
public void enterAnimate( )
```

- **Usage**

- \* Changes the user interface when the "Compile" button is pressed. Rewinds the animation.

---

- *enterEdit*

```
void enterEdit( )
```

- **Usage**

- \* This method is called when user clicks the "Edit" button.

---

- *enterEditTrue*

```
public void enterEditTrue( )
```

- **Usage**

- \* Makes the user interface changes when user clicks the "Edit" button.

---

- *findMainMethodCall*

```
public String findMainMethodCall( java.lang.String  programCode )
```

- **Usage**

- \* Tries to find the main method declaration from one of the classes.

---

- *freezeAnimation*

```
public void freezeAnimation( )
```

- **Usage**

- \* Changes the user interface when the animation is freezed.

---

- *getProgram*

```
public String getProgram( )
```

- **Usage**

- \* Returns the program code from the CodeEditor -object.

- **Returns** - The program code from the CodeEditor -object.

---

- *makeAnimationMenu*

```
private JMenu makeAnimationMenu( )
```

- **Usage**

- 
- \* Menu with the VCR commands
- 
- *makeControlButton*  
 private JButton **makeControlButton**( java.lang.String label,  
 java.lang.String iconName )
    - **Usage**
      - \* Makes the control buttons for the control panel.
    - **Parameters**
      - \* **label** - The label for the button.
      - \* **iconName** - The icon name for the icon on the button.
    - **Returns** - The control button for control panel.
- 
- *makeControlMenu*  
 private JMenu **makeControlMenu**( )
    - **Usage**
      - \* Menu with the commands to enter to animate and edit.
- 
- *makeControlPanel*  
 private JPanel **makeControlPanel**( )
    - **Usage**
      - \* Constructs the control panel. Uses makeControlButton(String, String)
    - **Returns** - The constructed control panel.
    - **See Also**
      - \* jeliot.gui.JeliotWindow.makeControlButton(String, String)
- 
- *makeHelpMenu*  
 private JMenu **makeHelpMenu**( )
    - **Usage**
      - \* Menu with the commands to enter to animate and edit.
- 
- *makeMenuBar*  
 private JMenuBar **makeMenuBar**( )
    - **Usage**
      - \* Makes and returns the menubar for the main frame. Things for debugging.
    - **Returns** - The menubar for the main frame.
- 
- *output*  
 public void **output**( java.lang.String str )
    - **Usage**
      - \* Writes the outputted string to the output console.
    - **Parameters**
      - \* **str** - String for output.
- 
- *pauseAnimation*  
 public void **pauseAnimation**( )

---

– **Usage**

\* Changes the user interface when the "Pause" button is pressed. Calls jeliot.pause() method.

– **See Also**

\* jeliot.Jeliot.pause()

---

• *playAnimation*

**void playAnimation( )**

– **Usage**

\* Changes the user interface when the "Play" button is pressed. Calls jeliot.play() method.

– **See Also**

\* jeliot.Jeliot.play()

---

• *removeComments*

**public String removeComments( java.lang.String programCode )**

– **Usage**

\* Removes the comments from the source code.

– **Parameters**

\* **programCode** - the source code

– **Returns** - the source code without comments

---

• *resumeAnimation*

**public void resumeAnimation( )**

– **Usage**

\* Changes the user interface when the "Resume" button is pressed.

---

• *rewindAnimation*

**void rewindAnimation( )**

– **Usage**

\* Changes the user interface when the "Rewind" button is pressed. Calls methods jeliot.rewind() and theatre.repaint().

– **See Also**

\* jeliot.Jeliot.rewind()

\* jeliot.theatre.Theatre.repaint()

---

• *runUntil*

**public void runUntil( )**

– **Usage**

\* Method is used to implement the run until feature.

---

• *runUntilDone*

**public void runUntilDone( )**

– **Usage**

\* Invoked when the runUntil is done.

---

- *setEnabledMenuItems*

```
public void setEnabledMenuItems( boolean   enabled, java.lang.String []
menuItems )
```

- **Usage**

- \* Sets the given menu items contained in the second parameter either enabled or disabled depending the value of the first parameter

- **Parameters**

- \* **enabled** - if true means that the given menu items should be enabled if false the menu items should be disabled.
  - \* **menuItems** - the menu items to be enabled or disabled.

---

- *setUp*

```
public void setUp( )
```

- **Usage**

- \* Initializes the JFrame frame. Sets up all the basic things for the window. (Panels, Panes, Menubars) Things for debugging.

---

- *showErrorMessage*

```
public void showErrorMessage( java.lang.Exception   e )
```

- **Usage**

- \* Show the error message of the exception in the theatre pane.

- **Parameters**

- \* **e** - The exception that is wanted to show.

---

- *showErrorMessage*

```
public void showErrorMessage( jeliot.mcode.InterpreterError   e )
```

- **Usage**

- \* Shows the error message and highlights the source code.

- **Parameters**

- \* **e** - interpreter error that contains the error message and the highlighting information.

- **See Also**

- \* `jeliot.gui.JeliotWindow.showErrorMessage(String)`

---

- *showErrorMessage*

```
public void showErrorMessage( java.lang.String   e )
```

- **Usage**

- \* Shows the given error message and sets the buttons and menuitems for animation as disabled.

- **Parameters**

- \* **e** - the error message in String

---

- *showMessagesInDialogs*

```
public boolean showMessagesInDialogs( )
```

- **Usage**

- \* Get the showMessagesInDialogs variables value

- **Returns -**

---

- *stepAnimation*

**void stepAnimation( )**

- **Usage**

- \* Changes the user interface when the "Step" button is pressed. Calls jeliot.step() method.

- **See Also**

- \* jeliot.Jeliot.step()

---

- *tryToEnterAnimate*

**void tryToEnterAnimate( )**

- **Usage**

- \* Called when the user pushes the "Compile" button. Gets the code from the CodeEditor -object. Sends it to "compilation".

### 6.1.8 CLASS LineNumbers

---

The LineNumbers component is used to show the line numbers in the scroll panes left side in the code view and code editor.

#### DECLARATION

---

<pre>public class LineNumbers <b>extends</b> javax.swing.JComponent</pre>
---

#### SERIALIZABLE FIELDS

---

- private int size
  - The width of the component.
- private Font font
  - The font for this component.
- private int ascent
  - The ascent of the font.
- private int increment
  - The increment between two lines.
- private Insets insets
  - insets in the component.

## FIELDS

---

- private int size
  - The width of the component.
- private Font font
  - The font for this component.
- private int ascent
  - The ascent of the font.
- private int increment
  - The increment between two lines.
- private Insets insets
  - insets in the component.

## CONSTRUCTORS

---

- *LineNumbers*  
**public LineNumbers( java.awt.Font font, java.awt.Insets insets )**
  - **Usage**
    - \* Sets the font and the insets and the determines the size increment and ascent from the font's font metrics.
  - **Parameters**
    - \* **font** - the font to be used in the component
    - \* **insets** - the insets for the layout.

## METHODS

---

- *paintComponent*  
**public void paintComponent( java.awt.Graphics g )**
- *setHeightByLines*  
**public void setHeightByLines( int lines )**
  - **Usage**
    - \* sets the height by the given number of lines that should be shown.
  - **Parameters**
    - \* **lines** -
- *setPreferredHeight*  
**public void setPreferredHeight( int ph )**
  - **Usage**
    - \* Sets the preferred height of the component.
  - **Parameters**
    - \* **ph** -

## METHODS INHERITED FROM CLASS javax.swing.JComponent

- 
- *\_paintImmediately*  
void **\_paintImmediately**( int , int , int , int )
  - *<clinit>*  
static void **<clinit>**( )
  - *addAncestorListener*  
public void **addAncestorListener**( javax.swing.event.AncestorListener )
  - *addNotify*  
public void **addNotify**( )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.beans.PropertyChangeListener )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.lang.String , java.beans.PropertyChangeListener )
  - *addVetoableChangeListener*  
public synchronized void **addVetoableChangeListener**( java.beans.VetoableChangeListener )
  - *adjustPaintFlags*  
private void **adjustPaintFlags**( )
  - *alwaysOnTop*  
boolean **alwaysOnTop**( )
  - *checkIfChildObscuredBySibling*  
boolean **checkIfChildObscuredBySibling**( )
  - *componentInputMapChanged*  
void **componentInputMapChanged**( javax.swing.ComponentInputMap )
  - *computeVisibleRect*  
static final void **computeVisibleRect**( java.awt.Component , java.awt.Rectangle )
  - *computeVisibleRect*  
public void **computeVisibleRect**( java.awt.Rectangle )
  - *compWriteObjectNotify*  
void **compWriteObjectNotify**( )
  - *contains*  
public boolean **contains**( int , int )
  - *createToolTip*  
public JToolTip **createToolTip**( )
  - *deregisterNextFocusableComponent*  
private void **deregisterNextFocusableComponent**( )
  - *disable*  
public void **disable**( )
  - *enable*  
public void **enable**( )
  - *enableSerialization*  
void **enableSerialization**( )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , boolean , boolean )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , byte , byte )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , char , char )

- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getAncestorListeners*  
public AncestorListener getAncestorListeners( )
- *getAutoscrolls*  
public boolean getAutoscrolls( )
- *getBorder*  
public Border getBorder( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary getClientProperties( )
- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )
- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )



- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )

- *getToolTipText*  
public String getToolTipText( )
- *getToolTipText*  
public String getToolTipText( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container getTopLevelAncestor( )
- *getTransferHandler*  
public TransferHandler getTransferHandler( )
- *getUIClassID*  
public String getUIClassID( )
- *getVerifyInputWhenFocusTarget*  
public boolean getVerifyInputWhenFocusTarget( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener getVetoableChangeListeners( )
- *getWidth*  
public int getWidth( )
- *getVisibleRect*  
public Rectangle getVisibleRect( )
- *getWriteObjCounter*  
static byte getWriteObjCounter( javax.swing.JComponent )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *grabFocus*  
public void grabFocus( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component )
- *isManagingFocus*  
public boolean isManagingFocus( )
- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )

- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )

- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener(  
java.beans.VetoableChangeListener )
- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )

- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )

- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )

- *adjustDecendantsOnParent*  
void **adjustDecendantsOnParent**( int    )
- *adjustDescendants*  
void **adjustDescendants**( int    )
- *adjustListeningChildren*  
void **adjustListeningChildren**( long    , int    )
- *applyComponentOrientation*  
public void **applyComponentOrientation**( java.awt.ComponentOrientation    )
- *areFocusTraversalKeysSet*  
public boolean **areFocusTraversalKeysSet**( int    )
- *checkGD*  
void **checkGD**( java.lang.String    )
- *clearCurrentFocusCycleRootOnHide*  
void **clearCurrentFocusCycleRootOnHide**(    )
- *clearMostRecentFocusOwnerOnHide*  
void **clearMostRecentFocusOwnerOnHide**(    )
- *containsFocus*  
final boolean **containsFocus**(    )
- *countComponents*  
public int **countComponents**(    )
- *countHierarchyMembers*  
int **countHierarchyMembers**(    )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int    , long    , boolean    )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int    , java.awt.Component    , java.awt.Container    , long    , boolean    )
- *deliverEvent*  
public void **deliverEvent**( java.awt.Event    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchEventToSelf*  
void **dispatchEventToSelf**( java.awt.AWTEvent    )
- *doLayout*  
public void **doLayout**(    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *findComponentAt*  
public Component **findComponentAt**( int    , int    )
- *findComponentAt*  
final Component **findComponentAt**( int    , int    , boolean    )
- *findComponentAt*  
public Component **findComponentAt**( java.awt.Point    )
- *findTraversalRoot*  
private Container **findTraversalRoot**(    )
- *getAccessibleAt*  
Accessible **getAccessibleAt**( java.awt.Point    )
- *getAccessibleChild*  
Accessible **getAccessibleChild**( int    )
- *getAccessibleChildrenCount*  
int **getAccessibleChildrenCount**(    )

- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getComponent*  
public Component **getComponent**( int )
- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentCount*  
public int **getComponentCount**( )
- *getComponents\_NoClientCode*  
final Component **getComponents\_NoClientCode**( )
- *getComponents*  
public Component **getComponents**( )
- *getContainerListeners*  
public synchronized ContainerListener **getContainerListeners**( )
- *getDropTargetEventTarget*  
Component **getDropTargetEventTarget**( int , int , boolean )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy **getFocusTraversalPolicy**( )
- *getInsets*  
public Insets **getInsets**( )
- *getLayout*  
public LayoutManager **getLayout**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )



- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )

- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )
- *removeAll*  
public void **removeAll**(    )
- *removeContainerListener*  
public synchronized void **removeContainerListener**( java.awt.event.ContainerListener    )
- *removeNotify*  
public void **removeNotify**(    )
- *setFocusCycleRoot*  
public void **setFocusCycleRoot**( boolean    )
- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int    , java.util.Set    )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy    )
- *setFont*  
public void **setFont**( java.awt.Font    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setZOrder*  
void **setZOrder**( java.awt.Component    , int    )
- *transferFocusBackward*  
public void **transferFocusBackward**(    )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**(    )
- *update*  
public void **update**( java.awt.Graphics    )
- *validate*  
public void **validate**(    )
- *validateTree*  
protected void **validateTree**(    )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )

METHODS INHERITED FROM CLASS `java.awt.Component`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *action*  
`public boolean action( java.awt.Event , java.lang.Object )`
  - *add*  
`public synchronized void add( java.awt.PopupMenu )`
  - *addComponentListener*  
`public synchronized void addComponentListener( java.awt.event.ComponentListener )`
  - *addFocusListener*  
`public synchronized void addFocusListener( java.awt.event.FocusListener )`
  - *addHierarchyBoundsListener*  
`public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )`
  - *addHierarchyListener*  
`public void addHierarchyListener( java.awt.event.HierarchyListener )`
  - *addInputMethodListener*  
`public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )`
  - *addKeyListener*  
`public synchronized void addKeyListener( java.awt.event.KeyListener )`
  - *addMouseListener*  
`public synchronized void addMouseListener( java.awt.event.MouseListener )`
  - *addMouseMotionListener*  
`public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )`
  - *addMouseWheelListener*  
`public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )`
  - *addNotify*  
`public void addNotify( )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
  - *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
  - *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
  - *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
  - *areInputMethodsEnabled*  
`boolean areInputMethodsEnabled( )`
  - *autoProcessMouseWheel*  
`void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )`
  - *autoTransferFocus*  
`final void autoTransferFocus( boolean )`

- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )

- *dispatchEvent*  
public final void **dispatchEvent**( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean **dispatchMouseWheelToAncestor**( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void **doAutoTransfer**( boolean    )
- *doLayout*  
public void **doLayout**( )
- *enable*  
public void **enable**( )
- *enable*  
public void **enable**( boolean    )
- *enableEvents*  
protected final void **enableEvents**( long    )
- *enableInputMethods*  
public void **enableInputMethods**( boolean    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean **eventTypeEnabled**( int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , int    , int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , java.lang.Object    ,  
java.lang.Object    )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getAccessibleIndexInParent*  
int **getAccessibleIndexInParent**( )
- *getAccessibleStateSet*  
AccessibleStateSet **getAccessibleStateSet**( )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle    )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )

- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )

- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener **getMouseMotionListeners**( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener **getMouseWheelListeners**( )
- *getName*  
public String **getName**( )
- *getNativeContainer*  
Container **getNativeContainer**( )
- *getParent\_NoClientCode*  
final Container **getParent\_NoClientCode**( )
- *getParent*  
public Container **getParent**( )
- *getPeer*  
public ComponentPeer **getPeer**( )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getSize*  
public Dimension **getSize**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getToolkitImpl*  
final Toolkit **getToolkitImpl**( )
- *getTreeLock*  
public final Object **getTreeLock**( )
- *getWidth*  
public int **getWidth**( )

- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *gotFocus*  
public boolean **gotFocus**( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean **handleEvent**( java.awt.Event )
- *hasFocus*  
public boolean **hasFocus**( )
- *hide*  
public void **hide**( )
- *imageUpdate*  
public boolean **imageUpdate**( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *inside*  
public boolean **inside**( int , int )
- *invalidate*  
public void **invalidate**( )
- *isBackgroundSet*  
public boolean **isBackgroundSet**( )
- *isCursorSet*  
public boolean **isCursorSet**( )
- *isDisplayable*  
public boolean **isDisplayable**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isEnabled*  
public boolean **isEnabled**( )
- *isEnabledImpl*  
final boolean **isEnabledImpl**( )
- *isFocusable*  
public boolean **isFocusable**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusOwner*  
public boolean **isFocusOwner**( )
- *isFocusTraversable*  
public boolean **isFocusTraversable**( )
- *isFocusTraversableOverridden*  
final boolean **isFocusTraversableOverridden**( )
- *isFontSet*  
public boolean **isFontSet**( )
- *isForegroundSet*  
public boolean **isForegroundSet**( )
- *isLightweight*  
public boolean **isLightweight**( )



- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )

- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )

- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *remove*  
public synchronized void remove( java.awt.MenuComponent    )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener    )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener    )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener    )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener    )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener    )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener    )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener    )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener    )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    , java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
protected boolean requestFocus( boolean    )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean    , boolean    )

- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )

- *setSize*  
public void **setSize**( java.awt.Dimension    )
- *setSize*  
public void **setSize**( int    , int    )
- *setVisible*  
public void **setVisible**( boolean    )
- *show*  
public void **show**(    )
- *show*  
public void **show**( boolean    )
- *size*  
public Dimension **size**(    )
- *toString*  
public String **toString**(    )
- *transferFocus*  
public void **transferFocus**(    )
- *transferFocusBackward*  
public void **transferFocusBackward**(    )
- *transferFocusUpCycle*  
public void **transferFocusUpCycle**(    )
- *update*  
public void **update**( java.awt.Graphics    )
- *updateCursorImmediately*  
final void **updateCursorImmediately**(    )
- *validate*  
public void **validate**(    )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )

### 6.1.9 CLASS LoadJeliot

---

This class is not used in the current version of Jeliot. LoadJeliot displays a splash screen and starts the Jeliot application.

#### DECLARATION

---

```
public class LoadJeliot
extends java.lang.Object
```

#### FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for gui package

CONSTRUCTORS

---

- *LoadJeliot*  
`public LoadJeliot( )`

METHODS

---

- *main*  
`public static void main( java.lang.String [] args )`
  - **Usage**  
 \* Starts the jeliot program.
  - **Parameters**  
 \* **args** - Command line arguments for the program.
  - **Exceptions**  
 \* `java.io.IOException` - If there is a problem in the opening of the file.

---
- *start*  
`public void start( )`
  - **Usage**  
 \* Initializes the Jeliot’s window. Initializes the `jeliot.Jeliot` object.

**6.1.10 CLASS OutputConsole**

---

`OutputConsole` is a text area on which the output of a user’s program is printed.

DECLARATION

---

```
public class OutputConsole
extends javax.swing.JTextArea
```

SERIALIZABLE FIELDS

---

- `public final JScrollPane` container
  - A scroll pane that contains the output console.
- `private Component` model
  - A component that is queried for the preferred and maximum height of the console.
- `private JPopupMenu` menu
  - The console’s popup menu has one choice for emptying the console.

## FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for gui package
- public final JScrollPane container
  - A scroll pane that contains the output console.
- private Component model
  - A component that is queried for the preferred and maximum height of the console.
- private JPopupMenu menu
  - The console's popup menu has one choice for emptying the console.

## CONSTRUCTORS

---

- *OutputConsole*  
`public OutputConsole( java.awt.Component model )`
  - **Usage**
    - \* Creates a new output console.
  - **Parameters**
    - \* **model** - The model is a component that is queried to set console's preferred and maximum height. May be null, in which case it has no effect (no error to be null).

## METHODS

---

- *isPopupTrigger*  
`private boolean isPopupTrigger( java.awt.event.MouseEvent evt )`
  - **Usage**
    - \* Checks if a mouse event should pop up the popup menu. A bit of a hack, because `InputEvent.isPopupTrigger()` doesn't seem to work on Windows95/jdk1.1.7a/swing1.1.1.
  - **Parameters**
    - \* **evt** - The mouse event that is supposed to be a popup menu trigger.
- *maybeShowPopup*  
`private void maybeShowPopup( java.awt.event.MouseEvent evt )`
  - **Usage**
    - \* Checks if a mouse click is a popup menu trigger and if it is, shows the popup menu.
  - **Parameters**
    - \* **evt** - The mouse event that is supposed to be a popup menu trigger.

## METHODS INHERITED FROM CLASS javax.swing.JTextArea

- 
- *append*  
public void **append**( java.lang.String )
  - *createDefaultModel*  
protected Document **createDefaultModel**( )
  - *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
  - *getColumns*  
public int **getColumns**( )
  - *getColumnWidth*  
protected int **getColumnWidth**( )
  - *getLineCount*  
public int **getLineCount**( )
  - *getLineEndOffset*  
public int **getLineEndOffset**( int )
  - *getLineOfOffset*  
public int **getLineOfOffset**( int )
  - *getLineStartOffset*  
public int **getLineStartOffset**( int )
  - *getLineWrap*  
public boolean **getLineWrap**( )
  - *getPreferredScrollableViewportSize*  
public Dimension **getPreferredScrollableViewportSize**( )
  - *getPreferredSize*  
public Dimension **getPreferredSize**( )
  - *getRowHeight*  
protected int **getRowHeight**( )
  - *getRows*  
public int **getRows**( )
  - *getScrollableTracksViewportWidth*  
public boolean **getScrollableTracksViewportWidth**( )
  - *getScrollableUnitIncrement*  
public int **getScrollableUnitIncrement**( java.awt.Rectangle , int , int )
  - *getTabSize*  
public int **getTabSize**( )
  - *getUIClassID*  
public String **getUIClassID**( )
  - *getWrapStyleWord*  
public boolean **getWrapStyleWord**( )
  - *insert*  
public void **insert**( java.lang.String , int )
  - *paramString*  
protected String **paramString**( )
  - *replaceRange*  
public void **replaceRange**( java.lang.String , int , int )
  - *setColumns*  
public void **setColumns**( int )
  - *setFont*  
public void **setFont**( java.awt.Font )



- *setLineWrap*  
public void setLineWrap( boolean )
- *setRows*  
public void setRows( int )
- *setTabSize*  
public void setTabSize( int )
- *setWrapStyleWord*  
public void setWrapStyleWord( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS javax.swing.text.JTextComponent

---

- *<clinit>*  
static void <clinit>( )
- *addCaretListener*  
public void addCaretListener( javax.swing.event.CaretListener )
- *addInputMethodListener*  
public void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeymap*  
public static Keymap addKeymap( java.lang.String , javax.swing.text.Keymap )
- *composedTextExists*  
boolean composedTextExists( )
- *copy*  
public void copy( )
- *createComposedString*  
private void createComposedString( int , java.text.AttributedCharacterIterator )
- *cut*  
public void cut( )
- *exchangeCaret*  
private void exchangeCaret( javax.swing.text.Caret , javax.swing.text.Caret )
- *fireCaretUpdate*  
protected void fireCaretUpdate( javax.swing.event.CaretEvent )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActions*  
public Action getActions( )
- *getCaret*  
public Caret getCaret( )
- *getCaretColor*  
public Color getCaretColor( )
- *getCaretListeners*  
public CaretListener getCaretListeners( )
- *getCaretPosition*  
public int getCaretPosition( )
- *getCurrentEventModifiers*  
private int getCurrentEventModifiers( )
- *getDisabledTextColor*  
public Color getDisabledTextColor( )

- *getDocument*  
public Document **getDocument**( )
- *getDragEnabled*  
public boolean **getDragEnabled**( )
- *getFocusAccelerator*  
public char **getFocusAccelerator**( )
- *getFocusedComponent*  
static final JTextComponent **getFocusedComponent**( )
- *getHighlighter*  
public Highlighter **getHighlighter**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeymap*  
public Keymap **getKeymap**( )
- *getKeymap*  
public static Keymap **getKeymap**( java.lang.String )
- *getMargin*  
public Insets **getMargin**( )
- *getNavigationFilter*  
public NavigationFilter **getNavigationFilter**( )
- *getPreferredScrollableViewportSize*  
public Dimension **getPreferredScrollableViewportSize**( )
- *getScrollableBlockIncrement*  
public int **getScrollableBlockIncrement**( java.awt.Rectangle , int , int )
- *getScrollableTracksViewportHeight*  
public boolean **getScrollableTracksViewportHeight**( )
- *getScrollableTracksViewportWidth*  
public boolean **getScrollableTracksViewportWidth**( )
- *getScrollableUnitIncrement*  
public int **getScrollableUnitIncrement**( java.awt.Rectangle , int , int )
- *getSelectedText*  
public String **getSelectedText**( )
- *getSelectedTextColor*  
public Color **getSelectedTextColor**( )
- *getSelectionColor*  
public Color **getSelectionColor**( )
- *getSelectionEnd*  
public int **getSelectionEnd**( )
- *getSelectionStart*  
public int **getSelectionStart**( )
- *getText*  
public String **getText**( )
- *getText*  
public String **getText**( int , int )
- *getToolTipText*  
public String **getToolTipText**( java.awt.event.MouseEvent )
- *getUI*  
public TextUI **getUI**( )
- *installDefaultTransferHandlerIfNecessary*  
private void **installDefaultTransferHandlerIfNecessary**( )

- *invokeAction*  
private void invokeAction( java.lang.String , javax.swing.Action )
- *isEditable*  
public boolean isEditable( )
- *isProcessInputMethodEventOverridden*  
private boolean isProcessInputMethodEventOverridden( )
- *isProcessInputMethodEventOverridden*  
private static Boolean isProcessInputMethodEventOverridden( java.lang.Class )
- *loadKeymap*  
public static void loadKeymap( javax.swing.text.Keymap ,  
javax.swing.text.JTextComponent.KeyBinding [] , javax.swing.Action [] )
- *mapCommittedTextToAction*  
private void mapCommittedTextToAction( java.lang.String )
- *modelToView*  
public Rectangle modelToView( int )
- *moveCaretPosition*  
public void moveCaretPosition( int )
- *paramString*  
protected String paramString( )
- *paste*  
public void paste( )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *read*  
public void read( java.io.Reader , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removeCaretListener*  
public void removeCaretListener( javax.swing.event.CaretListener )
- *removeKeymap*  
public static Keymap removeKeymap( java.lang.String )
- *removeNotify*  
public void removeNotify( )
- *replaceInputMethodText*  
private void replaceInputMethodText( java.awt.event.InputMethodEvent )
- *replaceSelection*  
public void replaceSelection( java.lang.String )
- *select*  
public void select( int , int )
- *selectAll*  
public void selectAll( )
- *setCaret*  
public void setCaret( javax.swing.text.Caret )
- *setCaretColor*  
public void setCaretColor( java.awt.Color )
- *setCaretPosition*  
public void setCaretPosition( int )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setDisabledTextColor*  
public void setDisabledTextColor( java.awt.Color )

- *setDocument*  
public void setDocument( javax.swing.text.Document    )
- *setDragEnabled*  
public void setDragEnabled( boolean    )
- *setEditable*  
public void setEditable( boolean    )
- *setFocusAccelerator*  
public void setFocusAccelerator( char    )
- *setHighlighter*  
public void setHighlighter( javax.swing.text.Highlighter    )
- *setInputMethodCaretPosition*  
private void setInputMethodCaretPosition( java.awt.event.InputMethodEvent    )
- *setKeymap*  
public void setKeymap( javax.swing.text.Keymap    )
- *setMargin*  
public void setMargin( java.awt.Insets    )
- *setNavigationFilter*  
public void setNavigationFilter( javax.swing.text.NavigationFilter    )
- *setSelectedTextColor*  
public void setSelectedTextColor( java.awt.Color    )
- *setSelectionColor*  
public void setSelectionColor( java.awt.Color    )
- *setSelectionEnd*  
public void setSelectionEnd( int    )
- *setSelectionStart*  
public void setSelectionStart( int    )
- *setText*  
public void setText( java.lang.String    )
- *setUI*  
public void setUI( javax.swing.plaf.TextUI    )
- *shouldSynthesizeKeyEvents*  
private boolean shouldSynthesizeKeyEvents(    )
- *updateInputMap*  
void updateInputMap( javax.swing.text.Keymap    , javax.swing.text.Keymap    )
- *updateUI*  
public void updateUI(    )
- *viewToModel*  
public int viewToModel( java.awt.Point    )
- *write*  
public void write( java.io.Writer    )

## METHODS INHERITED FROM CLASS javax.swing.JComponent

- 
- *\_paintImmediately*  
void **\_paintImmediately**( int , int , int , int )
  - *<clinit>*  
static void **<clinit>**( )
  - *addAncestorListener*  
public void **addAncestorListener**( javax.swing.event.AncestorListener )
  - *addNotify*  
public void **addNotify**( )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.beans.PropertyChangeListener )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.lang.String , java.beans.PropertyChangeListener )
  - *addVetoableChangeListener*  
public synchronized void **addVetoableChangeListener**( java.beans.VetoableChangeListener )
  - *adjustPaintFlags*  
private void **adjustPaintFlags**( )
  - *alwaysOnTop*  
boolean **alwaysOnTop**( )
  - *checkIfChildObscuredBySibling*  
boolean **checkIfChildObscuredBySibling**( )
  - *componentInputMapChanged*  
void **componentInputMapChanged**( javax.swing.ComponentInputMap )
  - *computeVisibleRect*  
static final void **computeVisibleRect**( java.awt.Component , java.awt.Rectangle )
  - *computeVisibleRect*  
public void **computeVisibleRect**( java.awt.Rectangle )
  - *compWriteObjectNotify*  
void **compWriteObjectNotify**( )
  - *contains*  
public boolean **contains**( int , int )
  - *createToolTip*  
public JToolTip **createToolTip**( )
  - *deregisterNextFocusableComponent*  
private void **deregisterNextFocusableComponent**( )
  - *disable*  
public void **disable**( )
  - *enable*  
public void **enable**( )
  - *enableSerialization*  
void **enableSerialization**( )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , boolean , boolean )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , byte , byte )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , char , char )

- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getAncestorListeners*  
public AncestorListener getAncestorListeners( )
- *getAutoscrolls*  
public boolean getAutoscrolls( )
- *getBorder*  
public Border getBorder( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary getClientProperties( )
- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )
- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )

- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )

- *getToolTipText*  
public String getToolTipText( )
- *getToolTipText*  
public String getToolTipText( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container getTopLevelAncestor( )
- *getTransferHandler*  
public TransferHandler getTransferHandler( )
- *getUIClassID*  
public String getUIClassID( )
- *getVerifyInputWhenFocusTarget*  
public boolean getVerifyInputWhenFocusTarget( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener getVetoableChangeListeners( )
- *getWidth*  
public int getWidth( )
- *getVisibleRect*  
public Rectangle getVisibleRect( )
- *getWriteObjCounter*  
static byte getWriteObjCounter( javax.swing.JComponent )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *grabFocus*  
public void grabFocus( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component )
- *isManagingFocus*  
public boolean isManagingFocus( )
- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )



- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )

- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener(  
java.beans.VetoableChangeListener )
- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )

- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )

- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )

- *adjustDecendantsOnParent*  
void **adjustDecendantsOnParent**( int    )
- *adjustDescendants*  
void **adjustDescendants**( int    )
- *adjustListeningChildren*  
void **adjustListeningChildren**( long    , int    )
- *applyComponentOrientation*  
public void **applyComponentOrientation**( java.awt.ComponentOrientation    )
- *areFocusTraversalKeysSet*  
public boolean **areFocusTraversalKeysSet**( int    )
- *checkGD*  
void **checkGD**( java.lang.String    )
- *clearCurrentFocusCycleRootOnHide*  
void **clearCurrentFocusCycleRootOnHide**( )
- *clearMostRecentFocusOwnerOnHide*  
void **clearMostRecentFocusOwnerOnHide**( )
- *containsFocus*  
final boolean **containsFocus**( )
- *countComponents*  
public int **countComponents**( )
- *countHierarchyMembers*  
int **countHierarchyMembers**( )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int    , long    , boolean    )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int    , java.awt.Component    , java.awt.Container    , long    , boolean    )
- *deliverEvent*  
public void **deliverEvent**( java.awt.Event    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchEventToSelf*  
void **dispatchEventToSelf**( java.awt.AWTEvent    )
- *doLayout*  
public void **doLayout**( )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *findComponentAt*  
public Component **findComponentAt**( int    , int    )
- *findComponentAt*  
final Component **findComponentAt**( int    , int    , boolean    )
- *findComponentAt*  
public Component **findComponentAt**( java.awt.Point    )
- *findTraversalRoot*  
private Container **findTraversalRoot**( )
- *getAccessibleAt*  
Accessible **getAccessibleAt**( java.awt.Point    )
- *getAccessibleChild*  
Accessible **getAccessibleChild**( int    )
- *getAccessibleChildrenCount*  
int **getAccessibleChildrenCount**( )

- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getComponent*  
public Component **getComponent**( int )
- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentCount*  
public int **getComponentCount**( )
- *getComponents\_NoClientCode*  
final Component **getComponents\_NoClientCode**( )
- *getComponents*  
public Component **getComponents**( )
- *getContainerListeners*  
public synchronized ContainerListener **getContainerListeners**( )
- *getDropTargetEventTarget*  
Component **getDropTargetEventTarget**( int , int , boolean )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy **getFocusTraversalPolicy**( )
- *getInsets*  
public Insets **getInsets**( )
- *getLayout*  
public LayoutManager **getLayout**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )

- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )

- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )
- *removeAll*  
public void **removeAll**(    )
- *removeContainerListener*  
public synchronized void **removeContainerListener**( java.awt.event.ContainerListener    )
- *removeNotify*  
public void **removeNotify**(    )
- *setFocusCycleRoot*  
public void **setFocusCycleRoot**( boolean    )
- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int    , java.util.Set    )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy    )
- *setFont*  
public void **setFont**( java.awt.Font    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setZOrder*  
void **setZOrder**( java.awt.Component    , int    )
- *transferFocusBackward*  
public void **transferFocusBackward**(    )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**(    )
- *update*  
public void **update**( java.awt.Graphics    )
- *validate*  
public void **validate**(    )
- *validateTree*  
protected void **validateTree**(    )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )



METHODS INHERITED FROM CLASS `java.awt.Component`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *action*  
`public boolean action( java.awt.Event , java.lang.Object )`
  - *add*  
`public synchronized void add( java.awt.PopupMenu )`
  - *addComponentListener*  
`public synchronized void addComponentListener( java.awt.event.ComponentListener )`
  - *addFocusListener*  
`public synchronized void addFocusListener( java.awt.event.FocusListener )`
  - *addHierarchyBoundsListener*  
`public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )`
  - *addHierarchyListener*  
`public void addHierarchyListener( java.awt.event.HierarchyListener )`
  - *addInputMethodListener*  
`public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )`
  - *addKeyListener*  
`public synchronized void addKeyListener( java.awt.event.KeyListener )`
  - *addMouseListener*  
`public synchronized void addMouseListener( java.awt.event.MouseListener )`
  - *addMouseMotionListener*  
`public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )`
  - *addMouseWheelListener*  
`public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )`
  - *addNotify*  
`public void addNotify( )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
  - *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
  - *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
  - *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
  - *areInputMethodsEnabled*  
`boolean areInputMethodsEnabled( )`
  - *autoProcessMouseWheel*  
`void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )`
  - *autoTransferFocus*  
`final void autoTransferFocus( boolean )`

- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )

- *dispatchEvent*  
public final void **dispatchEvent**( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean **dispatchMouseWheelToAncestor**( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void **doAutoTransfer**( boolean    )
- *doLayout*  
public void **doLayout**( )
- *enable*  
public void **enable**( )
- *enable*  
public void **enable**( boolean    )
- *enableEvents*  
protected final void **enableEvents**( long    )
- *enableInputMethods*  
public void **enableInputMethods**( boolean    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean **eventTypeEnabled**( int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , int    , int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , java.lang.Object    ,  
java.lang.Object    )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getAccessibleIndexInParent*  
int **getAccessibleIndexInParent**( )
- *getAccessibleStateSet*  
AccessibleStateSet **getAccessibleStateSet**( )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle    )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )

- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )

- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener **getMouseMotionListeners**( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener **getMouseWheelListeners**( )
- *getName*  
public String **getName**( )
- *getNativeContainer*  
Container **getNativeContainer**( )
- *getParent\_NoClientCode*  
final Container **getParent\_NoClientCode**( )
- *getParent*  
public Container **getParent**( )
- *getPeer*  
public ComponentPeer **getPeer**( )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getSize*  
public Dimension **getSize**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getToolkitImpl*  
final Toolkit **getToolkitImpl**( )
- *getTreeLock*  
public final Object **getTreeLock**( )
- *getWidth*  
public int **getWidth**( )

- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *gotFocus*  
public boolean **gotFocus**( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean **handleEvent**( java.awt.Event )
- *hasFocus*  
public boolean **hasFocus**( )
- *hide*  
public void **hide**( )
- *imageUpdate*  
public boolean **imageUpdate**( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *inside*  
public boolean **inside**( int , int )
- *invalidate*  
public void **invalidate**( )
- *isBackgroundSet*  
public boolean **isBackgroundSet**( )
- *isCursorSet*  
public boolean **isCursorSet**( )
- *isDisplayable*  
public boolean **isDisplayable**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isEnabled*  
public boolean **isEnabled**( )
- *isEnabledImpl*  
final boolean **isEnabledImpl**( )
- *isFocusable*  
public boolean **isFocusable**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusOwner*  
public boolean **isFocusOwner**( )
- *isFocusTraversable*  
public boolean **isFocusTraversable**( )
- *isFocusTraversableOverridden*  
final boolean **isFocusTraversableOverridden**( )
- *isFontSet*  
public boolean **isFontSet**( )
- *isForegroundSet*  
public boolean **isForegroundSet**( )
- *isLightweight*  
public boolean **isLightweight**( )

- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )

- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )



- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *repaint*  
public void repaint( )
- *repaint*  
public void repaint( int , int , int , int )
- *repaint*  
public void repaint( long )
- *repaint*  
public void repaint( long , int , int , int , int )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )

- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )

- *setSize*  
public void setSize( java.awt.Dimension    )
- *setSize*  
public void setSize( int    , int    )
- *setVisible*  
public void setVisible( boolean    )
- *show*  
public void show(    )
- *show*  
public void show( boolean    )
- *size*  
public Dimension size(    )
- *toString*  
public String toString(    )
- *transferFocus*  
public void transferFocus(    )
- *transferFocusBackward*  
public void transferFocusBackward(    )
- *transferFocusUpCycle*  
public void transferFocusUpCycle(    )
- *update*  
public void update( java.awt.Graphics    )
- *updateCursorImmediately*  
final void updateCursorImmediately(    )
- *validate*  
public void validate(    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

### 6.1.11 CLASS TheaterPopup

---

Currently this class is not used in the Jeliot. Theatre’s popup menu class that gives information about the actors fo the theatre.

#### DECLARATION

---

```
public class TheaterPopup
extends java.awt.event.MouseAdapter
implements java.awt.event.MouseMotionListener
```

#### CONSTRUCTORS

---

- *TheaterPopup*  
public **TheaterPopup**(    )

METHODS

---

• *handleMouseEvent*

```
private void handleMouseEvent( java.awt.event.MouseEvent evt )
```

– **Usage**

\* Method that handles the events when the actor should be highlighted.

– **Parameters**

\* **evt** - The mouse event that should be handled.

---

• *mouseDragged*

```
public void mouseDragged( java.awt.event.MouseEvent evt )
```

– **Usage**

\* Implemented to conform to MouseMotionListener interface.

---

• *mouseEntered*

```
public void mouseEntered( java.awt.event.MouseEvent evt )
```

– **Usage**

\* Method to handle the mouse events when the actors should be highlighted.

– **Parameters**

\* **evt** - The mouse event when mouse entered the area.

---

• *mouseExited*

```
public void mouseExited( java.awt.event.MouseEvent evt )
```

– **Usage**

\* Method to handle the mouse events when the actors should be unhighlighted.

– **Parameters**

\* **evt** - The mouse event when mouse exited the area.

---

• *mouseMoved*

```
public void mouseMoved( java.awt.event.MouseEvent evt )
```

– **Usage**

\* Method to handle the mouse events when the actors should be highlighted or unhighlighted.

– **Parameters**

\* **evt** - The mouse event when mouse is moved.

---

• *mousePressed*

```
public void mousePressed( java.awt.event.MouseEvent evt )
```

– **Usage**

\* Shows the popup menu when the button is pushed on some actor.

– **Parameters**

\* **evt** - The mouse event when mouse is pressed.

---

- *showPopup*

```
private void showPopup( java.awt.event.MouseEvent evt )
```

- **Usage**

- \* Method checks what kind of popup menu it should activate or should it activate any kind of popup menu.

- **Parameters**

- \* **evt** - The mouse event when mouse button is pressed.

METHODS INHERITED FROM CLASS `java.awt.event.MouseAdapter`

---

- *mouseClicked*

```
public void mouseClicked( java.awt.event.MouseEvent )
```

- *mouseEntered*

```
public void mouseEntered( java.awt.event.MouseEvent )
```

- *mouseExited*

```
public void mouseExited( java.awt.event.MouseEvent )
```

- *mousePressed*

```
public void mousePressed( java.awt.event.MouseEvent )
```

- *mouseReleased*

```
public void mouseReleased( java.awt.event.MouseEvent )
```

## Chapter 7

# Package koala.dynamicjava.interpreter

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Interfaces</b>	
<b>ClassLoaderContainer</b> .....	368
<i>The classes that implements this interface represent objects that contains a class loader</i>	
<b>Interpreter</b> .....	368
<i>The classes that implements this interface represent the objects that holds the objects needed for interpretation.</i>	
<b>Classes</b>	
<b>ClassFactory</b> .....	372
<i>The instances of this class dynamically create java classes</i>	
<b>ClassInfoCompiler</b> .....	383
<i>This class translates a class info into a Class object</i>	
<b>ClassInfoCompiler.ConstructorVisitor</b> .....	384
<i>To build the constructors</i>	
<b>ClassInfoCompiler.MembersVisitor</b> .....	408
<i>To visit the members of a type declaration</i>	
<b>ClassPool</b> .....	422
<i>The instances of this class contains classinfos</i>	
<b>EvaluationVisitor</b> .....	423
<i>This tree visitor evaluates each node of a syntax tree</i>	
<b>InterpreterException</b> .....	451
<i>This exception is thrown when an error append while interpreting a statement</i>	
<b>InterpreterException.SourceInformation</b> .....	453
<i>To represent the source code informations</i>	
<b>InterpreterUtilities</b> .....	454
<i>This class contains a collection of utility methods for interpretation</i>	
<b>InterpreterUtilities.AddOperation</b> .....	461
<i>To encapsulate +</i>	
<b>InterpreterUtilities.BinaryArithmeticOperation</b> .....	462
<i>To encapsulate a binary operator</i>	
<b>InterpreterUtilities.BinaryPredicate</b> .....	463
<i>To encapsulate a boolean binary operator</i>	
<b>InterpreterUtilities.BitAndOperation</b> .....	463
<i>To encapsulate &amp;</i>	
<b>InterpreterUtilities.BitOrOperation</b> .....	464

<i>To encapsulate —</i>	
<b>InterpreterUtilities.BitwiseOperation</b> .....	465
<i>To encapsulate a bitwise operator</i>	
<b>InterpreterUtilities.DivideOperation</b> .....	465
<i>To encapsulate /</i>	
<b>InterpreterUtilities.EqualToPredicate</b> .....	466
<i>To encapsulate ==</i>	
<b>InterpreterUtilities.GreaterOrEqualOperation</b> .....	467
<i>To encapsulate &gt;=</i>	
<b>InterpreterUtilities.GreaterThanOperation</b> .....	468
<i>To encapsulate &gt;</i>	
<b>InterpreterUtilities.LessOrEqualOperation</b> .....	468
<i>To encapsulate &lt;=</i>	
<b>InterpreterUtilities.LessThanOperation</b> .....	469
<i>To encapsulate &lt;</i>	
<b>InterpreterUtilities.MinusOperation</b> .....	469
<i>To encapsulate -</i>	
<b>InterpreterUtilities.MultiplyOperation</b> .....	470
<i>To encapsulate *</i>	
<b>InterpreterUtilities.NotEqualToPredicate</b> .....	471
<i>To encapsulate !=</i>	
<b>InterpreterUtilities.PlusOperation</b> .....	472
<i>To encapsulate +</i>	
<b>InterpreterUtilities.RelationalOperation</b> .....	473
<i>To encapsulate a relational operation</i>	
<b>InterpreterUtilities.RemainderOperation</b> .....	473
<i>To encapsulate %</i>	
<b>InterpreterUtilities.ShiftLeftOperation</b> .....	474
<i>To encapsulate &lt;&lt;</i>	
<b>InterpreterUtilities.ShiftOperation</b> .....	475
<i>To encapsulate a shift operator</i>	
<b>InterpreterUtilities.ShiftRightOperation</b> .....	475
<i>To encapsulate &gt;&gt;</i>	
<b>InterpreterUtilities.SubtractOperation</b> .....	476
<i>To encapsulate -</i>	
<b>InterpreterUtilities.UnaryOperation</b> .....	477
<i>To encapsulate an unary operator</i>	
<b>InterpreterUtilities.UnsignedShiftRightOperation</b> .....	478
<i>To encapsulate &gt;&gt;&gt;</i>	
<b>InterpreterUtilities.XOrOperation</b> .....	479
<i>To encapsulate ^</i>	
<b>Main</b> .....	479
<i>This file contains the entry point of the interpreter</i>	
<b>NameVisitor</b> .....	480
<i>This tree visitor resolves the ambiguity in identifiers in a syntax tree</i>	
<b>NodeProperties</b> .....	506
<i>This interface contains the names of the syntax tree properties defined by the interpretative kernel</i>	
<b>TreeClassFinder</b> .....	508

*The instances of the classes that implements this interface are used to find the fully qualified name of classes and to manage the loading of these classes.*

**TreeClassLoader** ..... 510  
*This class is responsible for loading bytecode classes*

*All classes *created* by TreeClassLoaders have an identical CodeSource .*

**TreeCompiler** ..... 515  
*This class contains methods to manage the creation of classes.*

**TreeCompiler.ClassInfoLoader** ..... 516  
*To load class infos instead of classes*

**TreeCompiler.CompilationUnitVisitor** ..... 519  
*To create the class infos for a compilation unit*

**TreeCompiler.PseudoError** ..... 533  
*To test the existance of a class without loading it*

**TreeInterpreter** ..... 535  
*This class contains method to interpret the constructs of the language.*

**TreeInterpreter.ConstructorParametersDescriptor** ..... 542  
*Used to store the informations about explicit constructors invocation*

**TreeInterpreter.MethodDescriptor** ..... 543  
*Used to store the informations about dynamically created methods*

**TypeChecker** ..... 543  
*This tree visitor checks the typing rules and loads the classes, fields and methods*

**UninitializedObject** ..... 571  
*The INSTANCE object of this class represents an uninitialized object*

---



## 7.1 Interfaces

### 7.1.1 INTERFACE **ClassLoaderContainer**

---

The classes that implements this interface represent objects that contains a class loader

#### DECLARATION

---

```
public interface ClassLoaderContainer
```

#### METHODS

---

- *getClassLoader*  
 public ClassLoader **getClassLoader**( )
  - **Usage**
    - \* Returns the class loader

### 7.1.2 INTERFACE **Interpreter**

---

The classes that implements this interface represent the objects that holds the objects needed for interpretation.

#### DECLARATION

---

```
public interface Interpreter
```

#### METHODS

---

- *addClassPath*  
 public void **addClassPath**( java.lang.String path )
  - **Usage**
    - \* Adds a class search path
  - **Parameters**
    - \* path - the path to add

---
- *addClassURL*  
 public void **addClassURL**( java.net.URL url )
  - **Usage**
    - \* Adds a class search URL
  - **Parameters**
    - \* url - the url to add

---

- *addLibraryPath*

`public void addLibraryPath( java.lang.String path )`

- **Usage**

- \* Adds a library search path

- **Parameters**

- \* path - the path to add

---

- *addLibrarySuffix*

`public void addLibrarySuffix( java.lang.String s )`

- **Usage**

- \* Adds a library file suffix

- **Parameters**

- \* s - the suffix to add

---

- *defineClass*

`public Class defineClass( java.lang.String name, byte [] code )`

- **Usage**

- \* Converts an array of bytes into an instance of class Class

- **Exceptions**

- \* java.lang.ClassFormatError - if the class could not be defined

---

- *defineVariable*

`public void defineVariable( java.lang.String name, java.lang.Object value )`

- **Usage**

- \* Defines a variable in the interpreter environment

- **Parameters**

- \* name - the variable's name

- \* value - the initial value of the variable

- **Exceptions**

- \* java.lang.IllegalStateException - if name is already defined

---

- *getAccessible*

`public boolean getAccessible( )`

- **Usage**

- \* Observe the state of calls to setAccessible()

---

- *getClassLoader*

`public ClassLoader getClassLoader( )`

- **Usage**

- \* Gets the class loader

---

- *getClassNames*

`public Set getClassNames( )`

- **Usage**
    - \* Returns the defined class names
  - **Returns** - a set of strings

---
- *getLibraryFinder*  
 public LibraryFinder **getLibraryFinder**( )  
  - **Usage**
    - \* Gets the library finder

---
- *getParserFactory*  
 public ParserFactory **getParserFactory**( )  
  - **Usage**
    - \* Gets the parser factory

---
- *getVariable*  
 public Object **getVariable**( java.lang.String name )  
  - **Usage**
    - \* Gets the value of a variable
  - **Parameters**
    - \* **name** - the variable's name
  - **Exceptions**
    - \* java.lang.IllegalStateException - if the variable do not exist

---
- *getVariableClass*  
 public Class **getVariableClass**( java.lang.String name )  
  - **Usage**
    - \* Gets the class of a variable
  - **Parameters**
    - \* **name** - the variable's name
  - **Exceptions**
    - \* java.lang.IllegalStateException - if the variable do not exist

---
- *getVariableNames*  
 public Set **getVariableNames**( )  
  - **Usage**
    - \* Returns the defined variable names
  - **Returns** - a set of strings

---
- *interpret*  
 public Object **interpret**( java.io.InputStream is, java.lang.String fname )  
  - **Usage**
    - \* Runs the interpreter
  - **Parameters**
    - \* **is** - the input stream from which the statements are read
    - \* **fname** - the name of the parsed stream

- **Returns** - the result of the evaluation of the last statement

---

- *interpret*

```
public Object interpret( java.io.Reader  r, java.lang.String  fname )
```

- **Usage**
  - \* Runs the interpreter
- **Parameters**
  - \* **r** - the reader
  - \* **fname** - the name of the parsed stream
- **Returns** - the result of the evaluation of the last statement

---

- *interpret*

```
public Object interpret( java.lang.String  fname )
```

- **Usage**
  - \* Runs the interpreter
- **Parameters**
  - \* **fname** - the name of a file to interpret
- **Returns** - the result of the evaluation of the last statement

---

- *loadClass*

```
public Class loadClass( java.lang.String  name )
```

- **Usage**
  - \* Loads an interpreted class
- **Parameters**
  - \* **s** - the fully qualified name of the class to load
- **Exceptions**
  - \* `java.lang.ClassNotFoundException` - if the class cannot be find

---

- *setAccessible*

```
public void setAccessible( boolean  accessible )
```

- **Usage**
  - \* Set the interpreter contexts to override public/protected/private access restrictions on the methods and fields it handles. Default should be false, i.e. enforce Java access. Setting to true should override this and allow access to all fields.

---

- *setVariable*

```
public void setVariable( java.lang.String  name, java.lang.Object  value )
```

- **Usage**
  - \* Sets the value a variable
- **Parameters**
  - \* **name** - the variable's name
  - \* **value** - the value of the variable
- **Exceptions**
  - \* `java.lang.IllegalStateException` - if the assignment is invalid

## 7.2 Classes

### 7.2.1 CLASS ClassFactory

---

The instances of this class dynamically create java classes

#### DECLARATION

---

```
public class ClassFactory
extends koala.dynamicjava.classfile.ClassFile
```

#### FIELDS

---

- private static final String VOID\_NAME  
—
- private static final String BOOLEAN\_NAME  
—
- private static final String CHAR\_NAME  
—
- private static final String INT\_NAME  
—
- private static final String LONG\_NAME  
—
- private static final String FLOAT\_NAME  
—
- private static final String DOUBLE\_NAME  
—
- private static final String BYTE\_NAME  
—
- private static final String SHORT\_NAME  
—
- private static final String OBJECT\_NAME  
—
- private static final String BOOLEAN\_WRAPPER\_NAME  
—
- private static final String CHAR\_WRAPPER\_NAME

- 
- private static final String INT\_WRAPPER\_NAME
- 
- private static final String LONG\_WRAPPER\_NAME
- 
- private static final String FLOAT\_WRAPPER\_NAME
- 
- private static final String DOUBLE\_WRAPPER\_NAME
- 
- private static final String BYTE\_WRAPPER\_NAME
- 
- private static final String SHORT\_WRAPPER\_NAME
- 
- private static final String NUMBER\_NAME
- 
- private static final ClassIdentifier BOOLEAN\_IDENTIFIER
  - The class "java.lang.Boolean" identifier
- private static final MethodIdentifier BOOLEAN\_CONSTRUCTOR
  - The Boolean(boolean) constructor identifier
- private static final ClassIdentifier CHARACTER\_IDENTIFIER
  - The class "java.lang.Character" identifier
- private static final MethodIdentifier CHARACTER\_CONSTRUCTOR
  - The Character(char) constructor identifier
- private static final ClassIdentifier INTEGER\_IDENTIFIER
  - The class "java.lang.Integer" identifier
- private static final MethodIdentifier INTEGER\_CONSTRUCTOR
  - The Integer(int) constructor identifier
- private static final ClassIdentifier LONG\_IDENTIFIER
  - The class "java.lang.Long" identifier
- private static final MethodIdentifier LONG\_CONSTRUCTOR
  - The Long(long) constructor identifier
- private static final ClassIdentifier FLOAT\_IDENTIFIER
  - The class "java.lang.Float" identifier
- private static final MethodIdentifier FLOAT\_CONSTRUCTOR

- The Float(float) constructor identifier
- private static final ClassIdentifier DOUBLE\_IDENTIFIER
  - The class "java.lang.Double" identifier
- private static final MethodIdentifier DOUBLE\_CONSTRUCTOR
  - The Double(double) constructor identifier
- private static final ClassIdentifier BYTE\_IDENTIFIER
  - The class "java.lang.Byte" identifier
- private static final MethodIdentifier BYTE\_CONSTRUCTOR
  - The Byte(byte) constructor identifier
- private static final ClassIdentifier SHORT\_IDENTIFIER
  - The class "java.lang.Short" identifier
- private static final MethodIdentifier SHORT\_CONSTRUCTOR
  - The Short(short) constructor identifier
- private static final ClassIdentifier OBJECT\_IDENTIFIER
  - The class "java.lang.Object" identifier
- private static final ClassIdentifier NUMBER\_IDENTIFIER
  - The class "java.lang.Number" identifier
- private static final MethodIdentifier BOOLEAN\_BOOLEAN\_VALUE\_METHOD
  - The identifier for Number.booleanValue()
- private static final MethodIdentifier CHARACTER\_CHAR\_VALUE\_METHOD
  - The identifier for Number.charValue()
- private static final MethodIdentifier NUMBER\_INT\_VALUE\_METHOD
  - The identifier for Number.intValue()
- private static final MethodIdentifier NUMBER\_LONG\_VALUE\_METHOD
  - The identifier for Number.longValue()
- private static final MethodIdentifier NUMBER\_FLOAT\_VALUE\_METHOD
  - The identifier for Number.floatValue()
- private static final MethodIdentifier NUMBER\_DOUBLE\_VALUE\_METHOD
  - The identifier for Number.doubleValue()
- private static final MethodIdentifier NUMBER\_BYTE\_VALUE\_METHOD
  - The identifier for Number.byteValue()
- private static final MethodIdentifier NUMBER\_SHORT\_VALUE\_METHOD
  - The identifier for Number.shortValue()
- private String name

- The name of the class to build
- private String superName
  - The name of the superclass of the class to build
- private boolean isStaticClass
  - Whether the class to build is static
- private MethodIdentifier interpreterMethod
  - The identifier of the method called from the generated methods
- private MethodIdentifier interpretArgumentsMethod
  - The identifier of the method called from the generated constructors
- private MethodIdentifier thrownExceptionMethod
  - The identifier for `ThrownException.getException()`
- private String thrownExceptionName
  - The name of the thrown exception
- private InnerClassesAttribute innerClassesAttribute
  - The `InnerClassAttribute`
- private String classLoaderId
  - The classloader identifier

## CONSTRUCTORS

---

- *ClassFactory*

```
public ClassFactory( int  af, java.lang.String  name, java.lang.String
sname, java.lang.Class  interp, java.lang.Class  except, java.lang.String
clid )
```

  - **Usage**
    - \* Creates a new class factory for an innerclass
  - **Parameters**
    - \* **af** - the access flags
    - \* **name** - the name of the class to create
    - \* **sname** - the name of the superclass
    - \* **interp** - the class of the interpreter used to evaluate the body of the methods. It must implement methods with the following signatures:
 

```
public static Object[] interpretArguments (String key, Object[] args)

public static Object invokeMethod (String key, Object obj, Object[]
args)
```
    - \* **except** - the class of the exception used to manage exception throwing. It must implement a method with the following signature:
 

```
public Throwable getException().
```
    - \* **clid** - the class loader identifier



METHODS

---

• *addConstantBooleanField*

```
public void addConstantBooleanField( int  af, java.lang.String  ft,  
java.lang.String  nm, java.lang.Boolean  v )
```

– **Usage**

\* Adds a constant boolean field to the class.

– **Parameters**

\* **af** - the access flags  
\* **ft** - the field type  
\* **nm** - the field name

---

• *addConstantDoubleField*

```
public void addConstantDoubleField( int  af, java.lang.String  ft,  
java.lang.String  nm, java.lang.Double  v )
```

– **Usage**

\* Adds a constant double field to the class.

– **Parameters**

\* **af** - the access flags  
\* **ft** - the field type  
\* **nm** - the field name

---

• *addConstantFloatField*

```
public void addConstantFloatField( int  af, java.lang.String  ft,  
java.lang.String  nm, java.lang.Float  v )
```

– **Usage**

\* Adds a constant float field to the class.

– **Parameters**

\* **af** - the access flags  
\* **ft** - the field type  
\* **nm** - the field name

---

• *addConstantIntField*

```
public void addConstantIntField( int  af, java.lang.String  ft,  
java.lang.String  nm, java.lang.Integer  v )
```

– **Usage**

\* Adds a constant int field to the class.

– **Parameters**

\* **af** - the access flags  
\* **ft** - the field type  
\* **nm** - the field name

---

• *addConstantLongField*

```
public void addConstantLongField( int  af, java.lang.String  ft,  
java.lang.String  nm, java.lang.Long  v )
```

– **Usage**

\* Adds a constant long field to the class.

– **Parameters**

\* **af** - the access flags  
 \* **ft** - the field type  
 \* **nm** - the field name

---

• *addConstantStringField*

```
public void addConstantStringField( int  af, java.lang.String  ft,
java.lang.String  nm, java.lang.String  v )
```

– **Usage**

\* Adds a constant string field to the class.

– **Parameters**

\* **af** - the access flags  
 \* **ft** - the field type  
 \* **nm** - the field name

---

• *addConstructor*

```
public String addConstructor( int  af, java.lang.String [] pt,
java.lang.String [] ex, java.lang.String  sup, java.lang.String [] st )
```

– **Usage**

\* Adds a constructor to the class. The body is automatically generated.

– **Parameters**

\* **af** - the access flags  
 \* **pt** - the parameters types  
 \* **ex** - the exceptions thrown  
 \* **sup** - the name of the super constructor  
 \* **st** - the initialization method parameter types

– **Returns** - the constructor identifier

---

• *addField*

```
public void addField( int  af, java.lang.String  ft, java.lang.String  nm )
```

– **Usage**

\* Adds a field to the class.

– **Parameters**

\* **af** - the access flags  
 \* **ft** - the field type  
 \* **nm** - the field name

---

• *addInnerClassesEntry*

```
public InnerClassesEntry addInnerClassesEntry( )
```

– **Usage**

\* Adds an InnerClasses entry

---

• *addMethod*

```
public String addMethod( int  af, java.lang.String  rt, java.lang.String
nm, java.lang.String [] pt, java.lang.String [] ex )
```

– **Usage**

- \* Adds a method to the class. The body is automatically generated.
  - **Parameters**
    - \* **af** - the access flags
    - \* **rt** - the return type
    - \* **nm** - the name of the method
    - \* **pt** - the parameters types
    - \* **ex** - the exceptions thrown
  - **Returns** - the method identifier
- 
- *addSuperMethodAccessor*

```
public void addSuperMethodAccessor( int  af, java.lang.String  rt,
java.lang.String  nm, java.lang.String [] pt, java.lang.String [] ex )
```

    - **Usage**
      - \* Adds a method that calls the supermethod of the given method
    - **Parameters**
      - \* **af** - the access flags
      - \* **rt** - the return type
      - \* **nm** - the name of the method
      - \* **pt** - the parameters types
      - \* **ex** - the exceptions thrown
- 
- *aload*

```
private void aload( int  local, java.io.DataOutputStream  data )
```

    - **Usage**
      - \* This method generates the code that loads a local variable on the stack
    - **Parameters**
      - \* **local** - the local variable to load
      - \* **data** - the stream where the code is written
- 
- *astore*

```
private void astore( int  local, java.io.DataOutputStream  data )
```

    - **Usage**
      - \* This method generates the code that stores a local variable from the stack
    - **Parameters**
      - \* **local** - the local variable to store
      - \* **data** - the stream where the code is written
- 
- *bipush*

```
private void bipush( int  cst, java.io.DataOutputStream  data )
```

    - **Usage**
      - \* This method generates the code that push a byte constant on the stack
    - **Parameters**
      - \* **cst** - the local variable to store
      - \* **data** - the stream where the code is written
- 
- *createClassInitializer*

```
public String createClassInitializer( )
```

- **Usage**
    - \* Creates the class initializer.
  - **Returns** - the method identifier

---
- *getByteCode*

```
public byte getByteCode( )
```

  - **Usage**
    - \* Returns the generated class

---
- *getMethodIdentifier*

```
public static String getMethodIdentifier( java.lang.String  cname,
java.lang.String  mname, java.lang.String []  pnames, java.lang.String
clid )
```

  - **Usage**
    - \* Computes a method identifier. It starts with the name of the class followed by a '#'.
  - **Parameters**
    - \* **cname** - the name of the class
    - \* **mname** - the name of the method
    - \* **pnames** - the names of the parameter classes
    - \* **clid** - the classloader identifier

---
- *iload*

```
private void iload( int  local, java.io.DataOutputStream  data )
```

  - **Usage**
    - \* This method generates the code that loads a local variable on the stack
  - **Parameters**
    - \* **local** - the local variable to load
    - \* **data** - the stream where the code is written

---
- *loadBoolean*

```
private void loadBoolean( int  currentLocal, java.io.DataOutputStream
data, koala.dynamicjava.classfile.ConstantPool  cp )
```

  - **Usage**
    - \* This method generates the code that puts a boolean argument in the table passed to 'Interpreter.invokeMethod'
  - **Parameters**
    - \* **currentLocal** - the current parameter
    - \* **data** - the stream where the code is written
    - \* **cp** - the constant pool

---
- *loadByte*

```
private void loadByte( int  currentLocal, java.io.DataOutputStream  data,
koala.dynamicjava.classfile.ConstantPool  cp )
```

  - **Usage**
    - \* This method generates the code that puts a byte argument in the table passed to 'Interpreter.invokeMethod'

---

– **Parameters**

- \* **currentLocal** - the current parameter
  - \* **data** - the stream where the code is written
  - \* **cp** - the constant pool
- 

• *loadChar*

```
private void loadChar( int    currentLocal, java.io.DataOutputStream data,
koala.dynamicjava.classfile.ConstantPool cp )
```

– **Usage**

- \* This method generates the code that puts a char argument in the table passed to 'Interpreter.invokeMethod'

– **Parameters**

- \* **currentLocal** - the current parameter
  - \* **data** - the stream where the code is written
  - \* **cp** - the constant pool
- 

• *loadDouble*

```
private void loadDouble( int    currentLocal, java.io.DataOutputStream data,
koala.dynamicjava.classfile.ConstantPool cp )
```

– **Usage**

- \* This method generates the code that puts a double argument in the table passed to 'Interpreter.invokeMethod'

– **Parameters**

- \* **currentLocal** - the current parameter
  - \* **data** - the stream where the code is written
  - \* **cp** - the constant pool
- 

• *loadFloat*

```
private void loadFloat( int    currentLocal, java.io.DataOutputStream data,
koala.dynamicjava.classfile.ConstantPool cp )
```

– **Usage**

- \* This method generates the code that puts a float argument in the table passed to 'Interpreter.invokeMethod'

– **Parameters**

- \* **currentLocal** - the current parameter
  - \* **data** - the stream where the code is written
  - \* **cp** - the constant pool
- 

• *loadInt*

```
private void loadInt( int    currentLocal, java.io.DataOutputStream data,
koala.dynamicjava.classfile.ConstantPool cp )
```

– **Usage**

- \* This method generates the code that puts an int argument in the table passed to 'Interpreter.invokeMethod'

– **Parameters**

- \* **currentLocal** - the current parameter
- \* **data** - the stream where the code is written

\* `cp` - the constant pool

---

- *loadLong*

```
private void loadLong( int    currentLocal, java.io.DataOutputStream data,
koala.dynamicjava.classfile.ConstantPool cp )
```

- **Usage**

- \* This method generates the code that puts a long argument in the table passed to 'Interpreter.invokeMethod'

- **Parameters**

- \* `currentLocal` - the current parameter
    - \* `data` - the stream where the code is written
    - \* `cp` - the constant pool

---

- *loadShort*

```
private void loadShort( int    currentLocal, java.io.DataOutputStream data,
koala.dynamicjava.classfile.ConstantPool cp )
```

- **Usage**

- \* This method generates the code that puts a short argument in the table passed to 'Interpreter.invokeMethod'

- **Parameters**

- \* `currentLocal` - the current parameter
    - \* `data` - the stream where the code is written
    - \* `cp` - the constant pool

## METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.ClassFile`

---

( in 14.1.5, page 1047)

- *addInterface*

```
public void addInterface( java.lang.String name )
```

- **Usage**

- \* Adds an interface to the list of the implemented interfaces

---

- *createField*

```
public FieldInfo createField( java.lang.String tp, java.lang.String nm )
```

- **Usage**

- \* Creates a new field

- **Parameters**

- \* `tp` - the type of the field
    - \* `nm` - the name of the field

- **See Also**

- \* `koala.dynamicjava.classfile.FieldInfo.FieldInfo`

---

- *createMethod*

```
public MethodInfo createMethod( java.lang.String rt, java.lang.String nm,
java.lang.String [] pt )
```

- **Usage**

- \* Creates a new method

- **Parameters**

- \* **rt** - the return type
  - \* **nm** - the name of the method
  - \* **pt** - the parameter types
- **See Also**
  - \* `koala.dynamicjava.classfile.MethodInfo.MethodInfo`

---

- *getConstantPool*  
**public ConstantPool getConstantPool( )**
  - **Usage**
    - \* Returns the constant pool

---

- *setAbstract*  
**public void setAbstract( )**
  - **Usage**
    - \* Sets the abstract flag for this class

---

- *setFinal*  
**public void setFinal( )**
  - **Usage**
    - \* Sets the final flag for this class

---

- *setInnerClassesAttribute*  
**public void setInnerClassesAttribute( koala.dynamicjava.classfile.InnerClassesAttribute attr )**
  - **Usage**
    - \* Sets the innerclasses attribute to the class
  - **Parameters**
    - \* **attr** - the attribute to set

---

- *setInterface*  
**public void setInterface( )**
  - **Usage**
    - \* Sets the interface flag for this class

---

- *setPublic*  
**public void setPublic( )**
  - **Usage**
    - \* Sets the public flag for this class

---

- *setSuper*  
**public void setSuper( )**
  - **Usage**
    - \* Sets the super flag for this class

---

- *write*  
**public void write( java.io.DataOutputStream out )**
  - **Usage**
    - \* Writes the class file to the given output stream

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.AttributeOwnerComponent`

---

( in 14.1.3, page 1045)

- *setAccessFlags*  
`public void setAccessFlags( int flags )`  
 – **Usage**  
   \* Sets the access flags for this class
- *setSyntheticAttribute*  
`public void setSyntheticAttribute( )`  
 – **Usage**  
   \* Sets the synthetic attribute to this field

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.BytecodeComponent`

---

( in 14.1.4, page 1046)

- *getConstantPool*  
`public ConstantPool getConstantPool( )`  
 – **Usage**  
   \* Returns the constant pool
- *write*  
`public abstract void write( java.io.DataOutputStream out )`  
 – **Usage**  
   \* Writes the class file to the given output stream
- *write*  
`public void write( java.io.OutputStream out )`  
 – **Usage**  
   \* Writes the class file to the given output stream

## 7.2.2 CLASS `ClassInfoCompiler`

---

This class translates a class info into a Class object

### DECLARATION

---

```
public class ClassInfoCompiler
  extends java.lang.Object
```

### CONSTRUCTORS

---

- *ClassInfoCompiler*  
`public ClassInfoCompiler( koala.dynamicjava.classinfo.ClassInfo ci )`  
 – **Usage**  
   \* Creates a new compiler  
 – **Parameters**  
   \* `ci` - the class info to compile



## METHODS

- *addConstructor*  
`protected void addConstructor(  
koala.dynamicjava.classinfo.TreeConstructorInfo ci )`
  - **Usage**  
\* Adds a constructor to the current class
  - **Parameters**  
\* ci - the constructor info

---

- *addInnerClassesAttribute*  
`protected void addInnerClassesAttribute(  
koala.dynamicjava.classinfo.ClassInfo ci )`
  - **Usage**  
\* Adds an inner class attribute to the given class

---

- *addToClassInitializer*  
`protected void addToClassInitializer( koala.dynamicjava.tree.Node n )`
  - **Usage**  
\* Adds a statement to the class initializer
  - **Parameters**  
\* n - the statement to add

---

- *addToInstanceInitializer*  
`protected void addToInstanceInitializer( koala.dynamicjava.tree.Node n )`
  - **Usage**  
\* Adds a statement to the instance initializer
  - **Parameters**  
\* n - the statement to add

---

- *compile*  
`public Class compile( )`
  - **Usage**  
\* Creates a Class object from the classInfo attribute
  - **Returns** - the created class

## 7.2.3 CLASS ClassInfoCompiler.ConstructorVisitor

To build the constructors

## DECLARATION

```
protected class ClassInfoCompiler.ConstructorVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

CONSTRUCTORS

---

- *ClassInfoCompiler.ConstructorVisitor*

protected **ClassInfoCompiler.ConstructorVisitor**( )

METHODS

---

- *checkList*

protected void **checkList**( java.util.List l, java.lang.String s,  
koala.dynamicjava.tree.Node n )

– **Usage**

\* Check a list of node

---

- *fieldExists*

protected boolean **fieldExists**( koala.dynamicjava.classinfo.ClassInfo dc,  
java.lang.String name )

– **Usage**

\* Whether the given name represents a field in this context

– **Parameters**

\* dc - the declaring class

\* name - the field name

---

- *visit*

public Object **visit**( koala.dynamicjava.tree.AddAssignExpression node )

– **Usage**

\* Visits an AddAssignExpression

– **Parameters**

\* node - the node to visit

---

- *visit*

public Object **visit**( koala.dynamicjava.tree.AddExpression node )

– **Usage**

\* Visits an AddExpression

– **Parameters**

\* node - the node to visit

---

- *visit*

public Object **visit**( koala.dynamicjava.tree.AndExpression node )

– **Usage**

\* Visits an AndExpression

– **Parameters**

\* node - the node to visit

---

- *visit*

public Object **visit**( koala.dynamicjava.tree.ArrayAccess node )

- **Usage**
    - \* Visits an `ArrayAccess`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

  - **Usage**
    - \* Visits an `ArrayAllocation`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayType node )
```

  - **Usage**
    - \* Visits a `ArrayType`
  - **Parameters**
    - \* `node` - the node to visit
  - **Returns** - the name of the type

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )
```

  - **Usage**
    - \* Visits a `BitAndAssignExpression`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndExpression node )
```

  - **Usage**
    - \* Visits a `BitAndExpression`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )
```

  - **Usage**
    - \* Visits a `BitOrAssignExpression`
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrExpression node )
```

  - **Usage**

- \* Visits a BitOrExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.CastExpression node )
  - **Usage**
  - \* Visits a CastExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.ComplementExpression node )
  - **Usage**
  - \* Visits a ComplementExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
  - **Usage**
  - \* Visits a ConditionalExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )
  - **Usage**
  - \* Visits a ConstructorInvocation
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
  - **Usage**
  - \* Visits an DivideAssignExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - public Object visit( koala.dynamicjava.tree.DivideExpression node )
  - **Usage**
  - \* Visits a DivideExpression
  - **Parameters**

\* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.EqualExpression node )

- **Usage**

- \* Visits an EqualExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )

- **Usage**

- \* Visits a ExclusiveOrAssignExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )

- **Usage**

- \* Visits a ExclusiveOrExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.FormalParameter node )

- **Usage**

- \* Visits a FormalParameter

- **Parameters**

- \* node - the node to visit

- **Returns** - the name of the parameter class

---

- *visit*

public Object visit( koala.dynamicjava.tree.GreaterExpression node )

- **Usage**

- \* Visits a GreaterExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )

- **Usage**

- \* Visits a GreaterOrEqualExpression

- **Parameters**

- 
- \* node - the node to visit

---

    - *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
```

      - **Usage**
        - \* Visits a InstanceOfExpression
      - **Parameters**
        - \* node - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.LessExpression node )
```

      - **Usage**
        - \* Visits a LessExpression
      - **Parameters**
        - \* node - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
```

      - **Usage**
        - \* Visits a LessOrEqualExpression
      - **Parameters**
        - \* node - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.Literal node )
```

      - **Usage**
        - \* Visits a Literal
      - **Parameters**
        - \* node - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.MinusExpression node )
```

      - **Usage**
        - \* Visits a MinusExpression
      - **Parameters**
        - \* node - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

      - **Usage**
        - \* Visits an MultiplyAssignExpression
      - **Parameters**
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyExpression node )  
 – **Usage**  
   \* Visits a MultiplyExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotEqualExpression node )  
 – **Usage**  
   \* Visits a NotEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotExpression node )  
 – **Usage**  
   \* Visits a NotExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )  
 – **Usage**  
   \* Visits an ObjectFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )  
 – **Usage**  
   \* Visits an ObjectMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.OrExpression node )  
 – **Usage**  
   \* Visits an OrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PlusExpression node )

- **Usage**
    - \* Visits a PlusExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement  node )
```

  - **Usage**
    - \* Visits a PostDecrement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement  node )
```

  - **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement  node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement  node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType  node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**
    - \* **node** - the node to visit
  - **Returns** - the name of the type

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.QualifiedName  node )
```

  - **Usage**



- \* Visits a QualifiedName
  - **Parameters**
    - \* **node** - the node to visit
  - **Returns** - a node that depends of the meaning of this name. It could be : a QualifiedName, a ReferenceType or a FieldAccess.

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReferenceType  node )
```

  - **Usage**
    - \* Visits a ReferenceType
  - **Parameters**
    - \* **node** - the node to visit
  - **Returns** - the name of the type

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression  node )
```

  - **Usage**
    - \* Visits an RemainderAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderExpression  node )
```

  - **Usage**
    - \* Visits a RemainderExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression  node )
```

  - **Usage**
    - \* Visits a ShiftLeftAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftExpression  node )
```

  - **Usage**
    - \* Visits a ShiftLeftExpression
  - **Parameters**
    - \* **node** - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression  node
)
```

  - **Usage**  
 \* Visits a ShiftRightAssignExpression
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ShiftRightExpression  node )
```

  - **Usage**  
 \* Visits a ShiftRightExpression
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.SimpleAllocation  node )
```

  - **Usage**  
 \* Visits a SimpleAllocation
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.SimpleAssignExpression  node )
```

  - **Usage**  
 \* Visits a SimpleAssignExpression
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.StaticFieldAccess  node )
```

  - **Usage**  
 \* Visits a StaticFieldAccess
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.StaticMethodCall  node )
```

  - **Usage**  
 \* Visits a StaticMethodCall
  - **Parameters**  
 \* `node` - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.SubtractAssignExpression  node )
```

- **Usage**
    - \* Visits an SubtractAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractExpression  node )
```

  - **Usage**
    - \* Visits a SubtractExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperFieldAccess  node )
```

  - **Usage**
    - \* Visits a SuperFieldAccess
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperMethodCall  node )
```

  - **Usage**
    - \* Visits a SuperMethodCall
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression  node )
```

  - **Usage**
    - \* Visits a ThisExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression  node )
```

  - **Usage**
    - \* Visits a TypeExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit(
koala.dynamicjava.tree.UnsignedShiftRightAssignExpression  node )
```

  - **Usage**

- \* Visits a UnsignedShiftRightAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression
node )
```

  - **Usage**
    - \* Visits a UnsignedShiftRightExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visitBinaryExpression*

```
protected void visitBinaryExpression(
koala.dynamicjava.tree.BinaryExpression node )
```

  - **Usage**
    - \* Visits the subexpressions of a BinaryExpression

---
- *visitBitwiseExpression*

```
protected void visitBitwiseExpression(
koala.dynamicjava.tree.BinaryExpression node, java.lang.String s )
```

  - **Usage**
    - \* Visits a bitwise expression

---
- *visitNumericExpression*

```
protected void visitNumericExpression(
koala.dynamicjava.tree.BinaryExpression node, java.lang.String s )
```

  - **Usage**
    - \* Visits a numeric expression

---
- *visitShiftExpression*

```
protected void visitShiftExpression( koala.dynamicjava.tree.BinaryExpression
node, java.lang.String s )
```

  - **Usage**
    - \* Visits a shift expression

---
- *visitUnaryExpression*

```
protected void visitUnaryExpression( koala.dynamicjava.tree.UnaryExpression
node )
```

  - **Usage**
    - \* Visits the subexpression of an UnaryExpression

---
- *visitUnaryOperation*

```
protected void visitUnaryOperation( koala.dynamicjava.tree.UnaryExpression
node, java.lang.String s )
```

  - **Usage**
    - \* Visits an unary operation

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.visitor.VisitorObject`

---

( in 5.2.1, page 160)

- *visit*  
`public Object visit( koala.dynamicjava.tree.AddAssignExpression node )`  
  - **Usage**  
\* Visits an AddAssignExpression
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.AddExpression node )`  
  - **Usage**  
\* Visits a AddExpression
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.AndExpression node )`  
  - **Usage**  
\* Visits a AndExpression
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayAccess node )`  
  - **Usage**  
\* Visits a ArrayAccess
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayAllocation node )`  
  - **Usage**  
\* Visits an ArrayAllocation
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayInitializer node )`  
  - **Usage**  
\* Visits an ArrayInitializer
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayType node )`  
  - **Usage**  
\* Visits a ArrayType
  - **Parameters**  
\* `node` - the node to visit

---

- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )  
 – **Usage**  
   \* Visits a BitAndAssignExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )  
 – **Usage**  
   \* Visits a BitOrExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )  
 – **Usage**  
   \* Visits a BlockStatement  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BreakStatement node )  
 – **Usage**  
   \* Visits a BreakStatement  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.CastExpression node )  
 – **Usage**  
   \* Visits a CastExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.CatchStatement node )

- **Usage**
    - \* Visits a CatchStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassAllocation node )
```

  - **Usage**
    - \* Visits an ClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration node )
```

  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer node )
```

  - **Usage**
    - \* Visits a ClassInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression node )
```

  - **Usage**
    - \* Visits a ComplementExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
```

  - **Usage**
    - \* Visits a ConditionalExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )
```

  - **Usage**
    - \* Visits a ConstructorDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )
```

  - **Usage**
    - \* Visits a ConstructorInvocation
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ContinueStatement node )  
 – **Usage**  
   \* Visits a ContinueStatement  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )  
 – **Usage**  
   \* Visits an DivideAssignExpression  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideExpression node )  
 – **Usage**  
   \* Visits a DivideExpression  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.DoStatement node )  
 – **Usage**  
   \* Visits a DoStatement  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.EmptyStatement node )  
 – **Usage**  
   \* Visits an EmptyStatement  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.EqualExpression node )  
 – **Usage**  
   \* Visits a EqualExpression  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )  
 – **Usage**  
   \* Visits a ExclusiveOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit
-



- *visit*  
 public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )  
 – **Usage**  
   \* Visits a ExclusiveOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FieldDeclaration node )  
 – **Usage**  
   \* Visits a FieldDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FormalParameter node )  
 – **Usage**  
   \* Visits a FormalParameter  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ForStatement node )  
 – **Usage**  
   \* Visits a ForStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FunctionCall node )  
 – **Usage**  
   \* Visits a FunctionCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterExpression node )  
 – **Usage**  
   \* Visits a GreaterExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )  
 – **Usage**  
   \* Visits a GreaterOrEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )

- **Usage**
    - \* Visits a IfThenElseStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenStatement  node )
```

  - **Usage**
    - \* Visits a IfThenStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration  node )
```

  - **Usage**
    - \* Visits an ImportDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerAllocation  node )
```

  - **Usage**
    - \* Visits an InnerAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerClassAllocation  node )
```

  - **Usage**
    - \* Visits an InnerClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceInitializer  node )
```

  - **Usage**
    - \* Visits a InstanceInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression  node )
```

  - **Usage**
    - \* Visits an InstanceOfExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration  node )
```

  - **Usage**
    - \* Visits a InterfaceDeclaration
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LabeledStatement node )  
 – Usage  
   \* Visits a LabeledStatement  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )  
 – Usage  
   \* Visits a LessExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )  
 – Usage  
   \* Visits a LessOrEqualExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.Literal node )  
 – Usage  
   \* Visits a Literal  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.MethodDeclaration node )  
 – Usage  
   \* Visits a MethodDeclaration  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.MinusExpression node )  
 – Usage  
   \* Visits a MinusExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )  
 – Usage  
   \* Visits an MultiplyAssignExpression  
 – Parameters  
   \* node - the node to visit
-

- *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyExpression node )  
 – **Usage**  
   \* Visits a MultiplyExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotEqualExpression node )  
 – **Usage**  
   \* Visits a NotEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotExpression node )  
 – **Usage**  
   \* Visits a NotExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )  
 – **Usage**  
   \* Visits a ObjectFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )  
 – **Usage**  
   \* Visits a ObjectMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.OrExpression node )  
 – **Usage**  
   \* Visits a OrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PackageDeclaration node )  
 – **Usage**  
   \* Visits an PackageDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PlusExpression node )

- **Usage**
    - \* Visits a PlusExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement  node )
```

  - **Usage**
    - \* Visits a PostDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement  node )
```

  - **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement  node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement  node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType  node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.QualifiedName  node )
```

  - **Usage**
    - \* Visits a QualifiedName
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReferenceType  node )
```

  - **Usage**
    - \* Visits a ReferenceType
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )  
 – Usage  
   \* Visits an RemainderAssignExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderExpression node )  
 – Usage  
   \* Visits a RemainderExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ReturnStatement node )  
 – Usage  
   \* Visits a ReturnStatement  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )  
 – Usage  
   \* Visits an ShiftLeftAssignExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )  
 – Usage  
   \* Visits a ShiftLeftExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )  
 – Usage  
   \* Visits an ShiftRightAssignExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )  
 – Usage  
   \* Visits a ShiftRightExpression  
 – Parameters  
   \* node - the node to visit
-

- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAllocation node )  
 – **Usage**  
   \* Visits an SimpleAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )  
 – **Usage**  
   \* Visits an SimpleAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )  
 – **Usage**  
   \* Visits a StaticFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticMethodCall node )  
 – **Usage**  
   \* Visits a StaticMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )  
 – **Usage**  
   \* Visits an SubtractAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractExpression node )  
 – **Usage**  
   \* Visits a SubtractExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )  
 – **Usage**  
   \* Visits a SuperFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperMethodCall node )  


---

- **Usage**
    - \* Visits a SuperMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchBlock  node )
```

  - **Usage**
    - \* Visits a SwitchBlock
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement  node )
```

  - **Usage**
    - \* Visits a SwitchStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement  node )
```

  - **Usage**
    - \* Visits a SynchronizedStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression  node )
```

  - **Usage**
    - \* Visits a ThisExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement  node )
```

  - **Usage**
    - \* Visits a ThrowStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement  node )
```

  - **Usage**
    - \* Visits a TryStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression  node )
```

  - **Usage**
    - \* Visits a TypeExpression
  - **Parameters**



- \* **node** - the node to visit
- • *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )  
 )
  - **Usage**
    - \* Visits an UnsignedShiftRightAssignExpression
  - **Parameters**
    - \* **node** - the node to visit
- • *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression node )  
 )
  - **Usage**
    - \* Visits a UnsignedShiftRightExpression
  - **Parameters**
    - \* **node** - the node to visit
- • *visit*  
 public Object visit( koala.dynamicjava.tree.VariableDeclaration node )  
 )
  - **Usage**
    - \* Visits a VariableDeclaration
  - **Parameters**
    - \* **node** - the node to visit
- • *visit*  
 public Object visit( koala.dynamicjava.tree.WhileStatement node )  
 )
  - **Usage**
    - \* Visits a WhileStatement
  - **Parameters**
    - \* **node** - the node to visit

## 7.2.4 CLASS ClassInfoCompiler.MembersVisitor

---

To visit the members of a type declaration

DECLARATION

---

```
protected class ClassInfoCompiler.MembersVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

CONSTRUCTORS

---

- *ClassInfoCompiler.MembersVisitor*  
 protected **ClassInfoCompiler.MembersVisitor**( )

METHODS

---

• *isRedefinedMethod*

```
protected boolean isRedefinedMethod( koala.dynamicjava.classinfo.MethodInfo
m )
```

– **Usage**

\* Whether the given method is a redefinition

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer node )
```

– **Usage**

\* Visits a ClassInitializer

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.FieldDeclaration node )
```

– **Usage**

\* Visits a FieldDeclaration

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceInitializer node )
```

– **Usage**

\* Visits a InstanceInitializer

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
```

– **Usage**

\* Visits a MethodDeclaration

– **Parameters**

\* **node** - the node to visit

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.visitor.VisitorObject`

---

( in 5.2.1, page 160)

• *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

– **Usage**

\* Visits an AddAssignExpression

– **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.AddExpression node )
      - Usage
        - \* Visits a AddExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.AndExpression node )
      - Usage
        - \* Visits a AndExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAccess node )
      - Usage
        - \* Visits a ArrayAccess
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
      - Usage
        - \* Visits an ArrayAllocation
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayInitializer node )
      - Usage
        - \* Visits an ArrayInitializer
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayType node )
      - Usage
        - \* Visits a ArrayType
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )
      - Usage
        - \* Visits a BitAndAssignExpression
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )  
 – **Usage**  
   \* Visits a BitOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )  
 – **Usage**  
   \* Visits a BlockStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BreakStatement node )  
 – **Usage**  
   \* Visits a BreakStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.CastExpression node )  
 – **Usage**  
   \* Visits a CastExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.CatchStatement node )  
 – **Usage**  
   \* Visits a CatchStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassAllocation node )

- **Usage**
    - \* Visits an ClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration  node )
```

  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer  node )
```

  - **Usage**
    - \* Visits a ClassInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression  node )
```

  - **Usage**
    - \* Visits a ComplementExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression  node )
```

  - **Usage**
    - \* Visits a ConditionalExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorDeclaration  node )
```

  - **Usage**
    - \* Visits a ConstructorDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorInvocation  node )
```

  - **Usage**
    - \* Visits a ConstructorInvocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ContinueStatement  node )
```

  - **Usage**
    - \* Visits a ContinueStatement
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*
    - public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
    - Usage
      - \* Visits an DivideAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.DivideExpression node )
    - Usage
      - \* Visits a DivideExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.DoStatement node )
    - Usage
      - \* Visits a DoStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.EmptyStatement node )
    - Usage
      - \* Visits an EmptyStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.EqualExpression node )
    - Usage
      - \* Visits a EqualExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
    - Usage
      - \* Visits a ExclusiveOrAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
    - Usage
      - \* Visits a ExclusiveOrExpression
      - Parameters
      - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.FieldDeclaration node )  
 – **Usage**  
   \* Visits a FieldDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FormalParameter node )  
 – **Usage**  
   \* Visits a FormalParameter  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ForStatement node )  
 – **Usage**  
   \* Visits a ForStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FunctionCall node )  
 – **Usage**  
   \* Visits a FunctionCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterExpression node )  
 – **Usage**  
   \* Visits a GreaterExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )  
 – **Usage**  
   \* Visits a GreaterOrEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )  
 – **Usage**  
   \* Visits a IfThenElseStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenStatement node )

- **Usage**
    - \* Visits a IfThenStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration node )
```

  - **Usage**
    - \* Visits an ImportDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerAllocation node )
```

  - **Usage**
    - \* Visits an InnerAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )
```

  - **Usage**
    - \* Visits an InnerClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceInitializer node )
```

  - **Usage**
    - \* Visits a InstanceInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
```

  - **Usage**
    - \* Visits an InstanceOfExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )
```

  - **Usage**
    - \* Visits a InterfaceDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.LabeledStatement node )
```

  - **Usage**
    - \* Visits a LabeledStatement
  - **Parameters**



- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )
      - Usage
        - \* Visits a LessExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
      - Usage
        - \* Visits a LessOrEqualExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.Literal node )
      - Usage
        - \* Visits a Literal
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
      - Usage
        - \* Visits a MethodDeclaration
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.MinusExpression node )
      - Usage
        - \* Visits a MinusExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
      - Usage
        - \* Visits an MultiplyAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
      - Usage
        - \* Visits a MultiplyExpression
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.NotEqualExpression node )  
 – **Usage**  
   \* Visits a NotEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotExpression node )  
 – **Usage**  
   \* Visits a NotExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )  
 – **Usage**  
   \* Visits a ObjectFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )  
 – **Usage**  
   \* Visits a ObjectMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.OrExpression node )  
 – **Usage**  
   \* Visits a OrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PackageDeclaration node )  
 – **Usage**  
   \* Visits an PackageDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PlusExpression node )  
 – **Usage**  
   \* Visits a PlusExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PostDecrement node )  


---

- **Usage**
    - \* Visits a PostDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement node )
```

  - **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.QualifiedName node )
```

  - **Usage**
    - \* Visits a QualifiedName
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReferenceType node )
```

  - **Usage**
    - \* Visits a ReferenceType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
```

  - **Usage**
    - \* Visits an RemainderAssignExpression
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderExpression node )
      - Usage
        - \* Visits a RemainderExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ReturnStatement node )
      - Usage
        - \* Visits a ReturnStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
      - Usage
        - \* Visits an ShiftLeftAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
      - Usage
        - \* Visits a ShiftLeftExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
      - Usage
        - \* Visits an ShiftRightAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
      - Usage
        - \* Visits a ShiftRightExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
      - Usage
        - \* Visits an SimpleAllocation
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )  
 – **Usage**  
   \* Visits an SimpleAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )  
 – **Usage**  
   \* Visits a StaticFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticMethodCall node )  
 – **Usage**  
   \* Visits a StaticMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )  
 – **Usage**  
   \* Visits an SubtractAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractExpression node )  
 – **Usage**  
   \* Visits a SubtractExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )  
 – **Usage**  
   \* Visits a SuperFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperMethodCall node )  
 – **Usage**  
   \* Visits a SuperMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SwitchBlock node )  


---

- **Usage**
    - \* Visits a SwitchBlock
  - **Parameters**
    - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement  node )
```

    - **Usage**
      - \* Visits a SwitchStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement  node )
```

    - **Usage**
      - \* Visits a SynchronizedStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression  node )
```

    - **Usage**
      - \* Visits a ThisExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement  node )
```

    - **Usage**
      - \* Visits a ThrowStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement  node )
```

    - **Usage**
      - \* Visits a TryStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression  node )
```

    - **Usage**
      - \* Visits a TypeExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression  node )
```

    - **Usage**
      - \* Visits an UnsignedShiftRightAssignExpression

- **Parameters**
    - \* *node* - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression  node )
```

    - **Usage**
      - \* Visits a UnsignedShiftRightExpression
    - **Parameters**
      - \* *node* - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration  node )
```

    - **Usage**
      - \* Visits a VariableDeclaration
    - **Parameters**
      - \* *node* - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.WhileStatement  node )
```

    - **Usage**
      - \* Visits a WhileStatement
    - **Parameters**
      - \* *node* - the node to visit

### 7.2.5 CLASS *ClassPool*

The instances of this class contains classinfos

#### DECLARATION

```
public class ClassPool
extends java.lang.Object
```

#### CONSTRUCTORS

- *ClassPool*

```
public ClassPool( )
```

#### METHODS

- *add*

```
public ClassInfo add( java.lang.String  cn,
koala.dynamicjava.classinfo.ClassInfo  ci )
```

  - **Usage**
    - \* Adds a classinfo to the pool
  - **Parameters**

- \* **cn** - the classname
  - \* **ci** - the classinfo
  - **Returns** - the given class info

---
- *contains*

```
public boolean contains( java.lang.String  cn )
```

  - **Usage**
    - \* Tests whether this pool contains the given class
  - **Parameters**
    - \* **cn** - the classname

---
- *get*

```
public ClassInfo get( java.lang.String  cn )
```

  - **Usage**
    - \* Returns the class info mapped with the given key
  - **Parameters**
    - \* **cn** - the classname

---
- *getFirstCompilable*

```
public ClassInfo getFirstCompilable( )
```

  - **Usage**
    - \* Gets the first compilable class in the pool
  - **Returns** - null if no class was found

## 7.2.6 CLASS *EvaluationVisitor*

---

This tree visitor evaluates each node of a syntax tree

### DECLARATION

---

```
public class EvaluationVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

### FIELDS

---

- private static long counter
  - Identifies each expression
- private boolean evaluating
  -
- private static boolean preparing
  -



- private static boolean inside
  -
- private static Stack constructorCallNames
  -
- private static Stack superClasses
  -
- private static Stack constructorCallNumbers
  -
- private static Stack returnExpressionCounterStack
  -
- private static Stack domesticStack
  -
- private List arrayCellNumbersList
  -
- private List arrayCellReferencesList
  -
- private Stack arrayCellNumbersStack
  -
- private Stack arrayCellReferencesStack
  -
- private Context context
  - The current context
- private boolean first
  - Visits an `ArrayAccess`

## CONSTRUCTORS

---

- *EvaluationVisitor*

```
public EvaluationVisitor( koala.dynamicjava.interpreter.context.Context  ctx
)
```

  - **Usage**
    - \* Creates a new visitor
  - **Parameters**
    - \* `ctx` - the current context

METHODS

---

- *constructorCallFinished*  
public static void constructorCallFinished( )
  - *display*  
public Object display( koala.dynamicjava.tree.QualifiedName node )
    - **Usage**
      - \* Displays a QualifiedName if it is declared without worrying about initialization
    - **Parameters**
      - \* **node** - the node to visit
    - **Returns** - the value of the local variable represented by this node
  - *extractSuperClasses*  
public Vector extractSuperClasses( java.lang.Class c )
  - *getConstructorCallName*  
public static String getConstructorCallName( )
  - *getConstructorCallNumber*  
public static long getConstructorCallNumber( )
  - *getCounter*  
public static long getCounter( )
  - *getSuperClasses*  
public static Vector getSuperClasses( )
  - *incrementCounter*  
public static void incrementCounter( )
  - *initialize*  
public void initialize( )
  - *isSetConstructorCall*  
public static boolean isSetConstructorCall( )
    - **Usage**
      - \* Returns constructorCall value
  - *isSetInside*  
public static boolean isSetInside( )
    - **Usage**
      - \* Returns preparing value
  - *isSetPreparing*  
public static boolean isSetPreparing( )
    - **Usage**
      - \* Returns preparing value
-

- *locationToString*

```
private String locationToString( koala.dynamicjava.tree.Node node )
```

- **Usage**

- \* Converts the node location into a string list. Each element is delimited by Code.LOC\_DELIM

- **Parameters**

- \* **node** - the node to visit

---

- *newConstructorCall*

```
public static void newConstructorCall( java.lang.String name,
java.util.Vector superClassesNames, long consCallNumber )
```

---

- *performCast*

```
private static Object performCast( java.lang.Class tc, java.lang.Object o
)
```

- **Usage**

- \* Performs a dynamic cast. This method acts on primitive wrappers.

- **Parameters**

- \* **tc** - the target class
    - \* **o** - the object to cast

---

- *setInside*

```
public static void setInside( )
```

- **Usage**

- \* Set inside value to true. So we have the information required (MD and PARAMETER) in a static method call

---

- *setPreparing*

```
public static void setPreparing( )
```

- **Usage**

- \* Set preparing value to true. So ArrayModifier is preparing the array to visit

---

- *unsetInside*

```
public static void unsetInside( )
```

- **Usage**

- \* Unset inside value to false.

---

- *unsetPreparing*

```
public static void unsetPreparing( )
```

- **Usage**

- \* Unset preparing value to false. So ArrayModifier has finished preparing the array to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

- **Usage**

- \* Visits an AddAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

  - **Usage**
    - \* Visits a AddExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

  - **Usage**
    - \* Visits an AndExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAccess node )
```

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

  - **Usage**
    - \* Visits an ArrayAllocation
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayInitializer node )
```

  - **Usage**
    - \* Visits a ArrayInitializer
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )
```

  - **Usage**
    - \* Visits a BitAndAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndExpression node )
```

  - **Usage**

- \* Visits a BitAndExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )`
  - **Usage**
  - \* Visits a BitOrAssignExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.BitOrExpression node )`
  - **Usage**
  - \* Visits a BitOrExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.BlockStatement node )`
  - **Usage**
  - \* Visits a BlockStatement
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.BreakStatement node )`
  - **Usage**
  - \* Visits a BreakStatement
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.CastExpression node )`
  - **Usage**
  - \* Visits a CastExpression
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.ClassAllocation node )`
  - **Usage**
  - \* Visits a ClassAllocation
  - **Parameters**

\* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ComplementExpression node )

- Usage

- \* Visits a ComplementExpression

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ConditionalExpression node )

- Usage

- \* Visits a ConditionalExpression

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ContinueStatement node )

- Usage

- \* Visits a ContinueStatement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )

- Usage

- \* Visits an DivideAssignExpression

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.DivideExpression node )

- Usage

- \* Visits a DivideExpression

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.DoStatement node )

- Usage

- \* Visits a DoStatement

- Parameters

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.EqualExpression node )
```

- **Usage**

- \* Visits an EqualExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
```

- **Usage**

- \* Visits a ExclusiveOrAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
```

- **Usage**

- \* Visits a ExclusiveOrExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ForStatement node )
```

- **Usage**

- \* Visits a ForStatement

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.FunctionCall node )
```

- **Usage**

- \* Visits a FunctionCall

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterExpression node )
```

- **Usage**

- \* Visits a GreaterExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )
```

- **Usage**
    - \* Visits a GreaterOrEqualExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )
```

  - **Usage**
    - \* Visits a IfThenElseStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenStatement node )
```

  - **Usage**
    - \* Visits a IfThenStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerAllocation node )
```

  - **Usage**
    - \* Visits a InnerAllocation
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
```

  - **Usage**
    - \* Visits a InstanceOfExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.LabeledStatement node )
```

  - **Usage**
    - \* Visits a LabeledStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.LessExpression node )
```

  - **Usage**
    - \* Visits a LessExpression



---

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )`

– **Usage**

\* Visits a LessOrEqualExpression

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.Literal node )`

– **Usage**

\* Visits a Literal

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.MinusExpression node )`

– **Usage**

\* Visits a MinusExpression

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )`

– **Usage**

\* Visits an MultiplyAssignExpression

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.MultiplyExpression node )`

– **Usage**

\* Visits a MultiplyExpression

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.NotEqualExpression node )`

– **Usage**

\* Visits a NotEqualExpression

– **Parameters**

\* **node** - the node to visit

---

- *visit*  
public Object visit( koala.dynamicjava.tree.NotExpression node )
  - Usage
    - \* Visits a NotExpression
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
  - Usage
    - \* Visits an ObjectFieldAccess
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
  - Usage
    - \* Visits an ObjectMethodCall
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.OrExpression node )
  - Usage
    - \* Visits an OrExpression
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.PlusExpression node )
  - Usage
    - \* Visits a PlusExpression
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.PostDecrement node )
  - Usage
    - \* Visits a PostDecrement
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.PostIncrement node )

- **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement  node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement  node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.QualifiedName  node )
```

  - **Usage**
    - \* Visits a QualifiedName
  - **Parameters**
    - \* **node** - the node to visit
  - **Returns** - the value of the local variable represented by this node

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression  node )
```

  - **Usage**
    - \* Visits an RemainderAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderExpression  node )
```

  - **Usage**
    - \* Visits a RemainderExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReturnStatement  node )
```

- **Usage**
    - \* Visits a ReturnStatement
  - **Parameters**
    - \* `node` - the node to visit
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
```

- **Usage**
    - \* Visits a ShiftLeftAssignExpression
  - **Parameters**
    - \* `node` - the node to visit
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
```

- **Usage**
    - \* Visits a ShiftLeftExpression
  - **Parameters**
    - \* `node` - the node to visit
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
```

- **Usage**
    - \* Visits a ShiftRightAssignExpression
  - **Parameters**
    - \* `node` - the node to visit
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
```

- **Usage**
    - \* Visits a ShiftRightExpression
  - **Parameters**
    - \* `node` - the node to visit
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
```

- **Usage**
    - \* Visits a SimpleAllocation
  - **Parameters**
    - \* `node` - the node to visit
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )
```

- **Usage**

- \* Visits a SimpleAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )
```

    - **Usage**
      - \* Visits a StaticFieldAccess
    - **Parameters**
      - \* **node** - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.StaticMethodCall node )
```

      - **Usage**
        - \* Visits a StaticMethodCall
      - **Parameters**
        - \* **node** - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
```

        - **Usage**
          - \* Visits an SubtractAssignExpression
        - **Parameters**
          - \* **node** - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractExpression node )
```

          - **Usage**
            - \* Visits a SubtractExpression
          - **Parameters**
            - \* **node** - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )
```

            - **Usage**
              - \* Visits a SuperFieldAccess
            - **Parameters**
              - \* **node** - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.SuperMethodCall node )
```

              - **Usage**
                - \* Visits a SuperMethodCall
              - **Parameters**

---

\* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.SwitchStatement node )

- Usage

- \* Visits a SwitchStatement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )

- Usage

- \* Visits a SynchronizedStatement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ThrowStatement node )

- Usage

- \* Visits a ThrowStatement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.TryStatement node )

- Usage

- \* Visits a TryStatement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.TypeExpression node )

- Usage

- \* Visits a TypeExpression

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit(  
koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )

- Usage

- \* Visits a UnsignedShiftRightAssignExpression

- Parameters

- \* node - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression
node )
```

  - **Usage**  
\* Visits a UnsignedShiftRightExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration node )
```

  - **Usage**  
\* Visits a VariableDeclaration
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.WhileStatement node )
```

  - **Usage**  
\* Visits a WhileStatement
  - **Parameters**  
\* node - the node to visit

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.visitor.VisitorObject

---

( in 5.2.1, page 160)

- *visit*  

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

  - **Usage**  
\* Visits an AddAssignExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

  - **Usage**  
\* Visits a AddExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

  - **Usage**  
\* Visits a AndExpression
  - **Parameters**  
\* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAccess node )  
 – **Usage**  
   \* Visits a ArrayAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAllocation node )  
 – **Usage**  
   \* Visits an ArrayAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayInitializer node )  
 – **Usage**  
   \* Visits an ArrayInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayType node )  
 – **Usage**  
   \* Visits a ArrayType  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )  
 – **Usage**  
   \* Visits a BitAndAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )



- **Usage**
    - \* Visits a BitOrExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BlockStatement  node )
```

  - **Usage**
    - \* Visits a BlockStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BreakStatement  node )
```

  - **Usage**
    - \* Visits a BreakStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.CastExpression  node )
```

  - **Usage**
    - \* Visits a CastExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.CatchStatement  node )
```

  - **Usage**
    - \* Visits a CatchStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassAllocation  node )
```

  - **Usage**
    - \* Visits an ClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration  node )
```

  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer  node )
```

  - **Usage**
    - \* Visits a ClassInitializer
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ComplementExpression node )  
 – Usage  
   \* Visits a ComplementExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConditionalExpression node )  
 – Usage  
   \* Visits a ConditionalExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )  
 – Usage  
   \* Visits a ConstructorDeclaration  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )  
 – Usage  
   \* Visits a ConstructorInvocation  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ContinueStatement node )  
 – Usage  
   \* Visits a ContinueStatement  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )  
 – Usage  
   \* Visits an DivideAssignExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideExpression node )  
 – Usage  
   \* Visits a DivideExpression  
 – Parameters  
   \* node - the node to visit
-

- *visit*  
 public Object visit( koala.dynamicjava.tree.DoStatement node )  
 – **Usage**  
   \* Visits a DoStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.EmptyStatement node )  
 – **Usage**  
   \* Visits an EmptyStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.EqualExpression node )  
 – **Usage**  
   \* Visits a EqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )  
 – **Usage**  
   \* Visits a ExclusiveOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )  
 – **Usage**  
   \* Visits a ExclusiveOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FieldDeclaration node )  
 – **Usage**  
   \* Visits a FieldDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FormalParameter node )  
 – **Usage**  
   \* Visits a FormalParameter  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ForStatement node )

- **Usage**
    - \* Visits a ForStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.FunctionCall node )
```

  - **Usage**
    - \* Visits a FunctionCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterExpression node )
```

  - **Usage**
    - \* Visits a GreaterExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )
```

  - **Usage**
    - \* Visits a GreaterOrEqualExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )
```

  - **Usage**
    - \* Visits a IfThenElseStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenStatement node )
```

  - **Usage**
    - \* Visits a IfThenStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration node )
```

  - **Usage**
    - \* Visits an ImportDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerAllocation node )
```

  - **Usage**
    - \* Visits an InnerAllocation
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )  
 – Usage  
   \* Visits an InnerClassAllocation  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceInitializer node )  
 – Usage  
   \* Visits a InstanceInitializer  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )  
 – Usage  
   \* Visits an InstanceOfExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )  
 – Usage  
   \* Visits a InterfaceDeclaration  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LabeledStatement node )  
 – Usage  
   \* Visits a LabeledStatement  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )  
 – Usage  
   \* Visits a LessExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )  
 – Usage  
   \* Visits a LessOrEqualExpression  
 – Parameters  
   \* node - the node to visit
-

- *visit*  
 public Object visit( koala.dynamicjava.tree.Literal node )  
 – **Usage**  
   \* Visits a Literal  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.MethodDeclaration node )  
 – **Usage**  
   \* Visits a MethodDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.MinusExpression node )  
 – **Usage**  
   \* Visits a MinusExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )  
 – **Usage**  
   \* Visits an MultiplyAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyExpression node )  
 – **Usage**  
   \* Visits a MultiplyExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotEqualExpression node )  
 – **Usage**  
   \* Visits a NotEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotExpression node )  
 – **Usage**  
   \* Visits a NotExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )

- **Usage**
    - \* Visits a ObjectFieldAccess
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall  node )
```

  - **Usage**
    - \* Visits a ObjectMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.OrExpression  node )
```

  - **Usage**
    - \* Visits a OrExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration  node )
```

  - **Usage**
    - \* Visits an PackageDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PlusExpression  node )
```

  - **Usage**
    - \* Visits a PlusExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement  node )
```

  - **Usage**
    - \* Visits a PostDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement  node )
```

  - **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement  node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.PreIncrement node )
      - Usage
        - \* Visits a PreIncrement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.PrimitiveType node )
      - Usage
        - \* Visits a PrimitiveType
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.QualifiedName node )
      - Usage
        - \* Visits a QualifiedName
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ReferenceType node )
      - Usage
        - \* Visits a ReferenceType
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
      - Usage
        - \* Visits an RemainderAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderExpression node )
      - Usage
        - \* Visits a RemainderExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ReturnStatement node )
      - Usage
        - \* Visits a ReturnStatement
      - Parameters
        - \* node - the node to visit

---



- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )  
 – **Usage**  
   \* Visits an ShiftLeftAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )  
 – **Usage**  
   \* Visits a ShiftLeftExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )  
 – **Usage**  
   \* Visits an ShiftRightAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )  
 – **Usage**  
   \* Visits a ShiftRightExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAllocation node )  
 – **Usage**  
   \* Visits an SimpleAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )  
 – **Usage**  
   \* Visits an SimpleAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )  
 – **Usage**  
   \* Visits a StaticFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticMethodCall node )

- **Usage**
    - \* Visits a StaticMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
```

  - **Usage**
    - \* Visits an SubtractAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractExpression node )
```

  - **Usage**
    - \* Visits a SubtractExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )
```

  - **Usage**
    - \* Visits a SuperFieldAccess
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperMethodCall node )
```

  - **Usage**
    - \* Visits a SuperMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchBlock node )
```

  - **Usage**
    - \* Visits a SwitchBlock
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement node )
```

  - **Usage**
    - \* Visits a SwitchStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )
```

  - **Usage**
    - \* Visits a SynchronizedStatement
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ThisExpression node )  
 – Usage  
   \* Visits a ThisExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ThrowStatement node )  
 – Usage  
   \* Visits a ThrowStatement  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.TryStatement node )  
 – Usage  
   \* Visits a TryStatement  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.TypeExpression node )  
 – Usage  
   \* Visits a TypeExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )  
 – Usage  
   \* Visits an UnsignedShiftRightAssignExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression node )  
 – Usage  
   \* Visits a UnsignedShiftRightExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.VariableDeclaration node )  
 – Usage  
   \* Visits a VariableDeclaration  
 – Parameters  
   \* node - the node to visit
-

- *visit*  
 public Object visit( koala.dynamicjava.tree.WhileStatement node )
  - **Usage**  
 \* Visits a WhileStatement
  - **Parameters**  
 \* node - the node to visit

## 7.2.7 CLASS InterpreterException

---

This exception is thrown when an error append while interpreting a statement

### DECLARATION

---

```
public class InterpreterException
extends koala.dynamicjava.interpreter.throwable.ThrownException
```

### CONSTRUCTORS

---

- *InterpreterException*  
 public **InterpreterException**(  
 koala.dynamicjava.interpreter.error.ExecutionError e )
  - **Usage**  
 \* Constructs an InterpreterException from a ExecutionError
- *InterpreterException*  
 public **InterpreterException**( koala.dynamicjava.parser.wrapper.ParseError e  
 )
  - **Usage**  
 \* Constructs an InterpreterException from a ParseError

### METHODS

---

- *getError*  
 public Throwable **getError**( )
- *getMessage*  
 public String **getMessage**( )
  - **Usage**  
 \* Returns the detailed message
- *getSourceInformation*  
 public InterpreterException.SourceInformation **getSourceInformation**( )
  - **Usage**  
 \* Returns the source code information if available, or null

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.throwable.ThrownException

---

( in 10.1.4, page 644)

- *getException*  
 public Throwable **getException**( )
  - **Usage**
    - \* Returns the exception that causes this error throwing

METHODS INHERITED FROM CLASS koala.dynamicjava.interpreter.error.ExecutionError

---

( in 17.1.2, page 1332)

- *getMessage*  
 public String **getMessage**( )
  - **Usage**
    - \* Returns the error message string of this exception
- *getNode*  
 public Node **getNode**( )
  - **Usage**
    - \* Returns the syntax tree node where the error occurs
- *printStackTrace*  
 public void **printStackTrace**( )
  - **Usage**
    - \* Overridden to delegate to printStackTrace(PrintStream) to print nested exception information.
  - **See Also**
    - \*  
 koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintStream)
- *printStackTrace*  
 public void **printStackTrace**( java.io.PrintStream s )
  - **Usage**
    - \* Overridden to delegate to printStackTrace(PrintWriter) to print nested exception information.
  - **See Also**
    - \*  
 koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintWriter)
- *printStackTrace*  
 public void **printStackTrace**( java.io.PrintWriter w )
  - **Usage**
    - \* Handles all calls to printStackTrace(), printing the stack trace of the current exception, and also that of its cause.

METHODS INHERITED FROM CLASS java.lang.Error

---

METHODS INHERITED FROM CLASS `java.lang.Throwable`

- 
- *fillInStackTrace*  
`public synchronized native Throwable fillInStackTrace( )`
  - *getCause*  
`public Throwable getCause( )`
  - *getLocalizedMessage*  
`public String getLocalizedMessage( )`
  - *getMessage*  
`public String getMessage( )`
  - *getOurStackTrace*  
`private synchronized StackTraceElement getOurStackTrace( )`
  - *getStackTrace*  
`public StackTraceElement getStackTrace( )`
  - *getStackTraceDepth*  
`private native int getStackTraceDepth( )`
  - *getStackTraceElement*  
`private native StackTraceElement getStackTraceElement( int )`
  - *initCause*  
`public synchronized Throwable initCause( java.lang.Throwable )`
  - *printStackTrace*  
`public void printStackTrace( )`
  - *printStackTrace*  
`public void printStackTrace( java.io.PrintStream )`
  - *printStackTrace*  
`public void printStackTrace( java.io.PrintWriter )`
  - *printStackTraceAsCause*  
`private void printStackTraceAsCause( java.io.PrintStream , java.lang.StackTraceElement [] )`
  - *printStackTraceAsCause*  
`private void printStackTraceAsCause( java.io.PrintWriter , java.lang.StackTraceElement [] )`
  - *setStackTrace*  
`public void setStackTrace( java.lang.StackTraceElement [] )`
  - *toString*  
`public String toString( )`
  - *writeObject*  
`private synchronized void writeObject( java.io.ObjectOutputStream )`

**7.2.8 CLASS `InterpreterException.SourceInformation`**


---

To represent the source code informations

## DECLARATION

---

```
public static class InterpreterException.SourceInformation
extends java.lang.Object
```

## FIELDS

---

- private String filename

–

- private int line

–

- private int column

–

## CONSTRUCTORS

---

- *InterpreterException.SourceInformation*

```
public InterpreterException.SourceInformation( java.lang.String  filename,  
int  line, int  column )
```

– **Usage**

\* Creates a source information

## METHODS

---

- *getColumn*

```
public int getColumn( )
```

– **Usage**

\* Returns the column where the error occurs

- 
- *getFilename*

```
public String getFilename( )
```

– **Usage**

\* Returns the filename

- 
- *getLine*

```
public int getLine( )
```

– **Usage**

\* Returns the line where the error occurs

## 7.2.9 CLASS **InterpreterUtilities**

---

This class contains a collection of utility methods for interpretation

DECLARATION

---

```
public class InterpreterUtilities
extends java.lang.Object
```

FIELDS

---

- public static final Byte ONE

–

CONSTRUCTORS

---

- *InterpreterUtilities*

```
protected InterpreterUtilities( )
```

– **Usage**

\* This class contains only static methods, so it is not useful to create instances of it.

METHODS

---

- *add*

```
public static Object add( java.lang.Class c, java.lang.Object l,
java.lang.Object r )
```

– **Usage**

\* Returns the value of an addition

– **Parameters**

\* **c** - the class of the result  
 \* **l** - the left operand  
 \* **r** - the right operand

- 
- *binaryArithmeticOperation*

```
protected static Object binaryArithmeticOperation( java.lang.Class c,
java.lang.Object l, java.lang.Object r,
koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation
o )
```

– **Usage**

\* Returns the value of a binary arithmetic operation

– **Parameters**

\* **c** - the class of the result  
 \* **l** - the left operand  
 \* **r** - the right operand  
 \* **o** - the operation

---



- *bitAnd*

```
public static Object bitAnd( java.lang.Class  c, java.lang.Object  l,
java.lang.Object  r )
```

- **Usage**

- \* Returns the value of a bit and operation

- **Parameters**

- \* *c* - the class of the result
  - \* *l* - the left operand
  - \* *r* - the right operand

---

- *bitOr*

```
public static Object bitOr( java.lang.Class  c, java.lang.Object  l,
java.lang.Object  r )
```

- **Usage**

- \* Returns the value of a bit or operation

- **Parameters**

- \* *c* - the class of the result
  - \* *l* - the left operand
  - \* *r* - the right operand

---

- *bitwiseOperation*

```
protected static Object bitwiseOperation( java.lang.Class  c,
java.lang.Object  l, java.lang.Object  r,
koala.dynamicjava.interpreter.InterpreterUtilities.BitwiseOperation  o )
```

- **Usage**

- \* Returns the value of a bitwise operation

- **Parameters**

- \* *c* - the class of the result
  - \* *l* - the left operand
  - \* *r* - the right operand
  - \* *o* - the operation

---

- *divide*

```
public static Object divide( java.lang.Class  c, java.lang.Object  l,
java.lang.Object  r )
```

- **Usage**

- \* Returns the value of a division

- **Parameters**

- \* *c* - the class of the result
  - \* *l* - the left operand
  - \* *r* - the right operand

---

- *equalityOperation*

```
protected static Object equalityOperation( java.lang.Class  lc,
java.lang.Class  rc, java.lang.Object  l, java.lang.Object  r,
koala.dynamicjava.interpreter.InterpreterUtilities.BinaryPredicate  p )
```

- **Usage**

- \* Returns the value of an equality operation
  - **Parameters**
    - \* **lc** - the class of the left operand
    - \* **rc** - the class of the right operand
    - \* **l** - the left operand
    - \* **r** - the right operand
    - \* **p** - the predicate to use

---
- *equalTo*

```
public static Object equalTo( java.lang.Class lc, java.lang.Class rc,
                             java.lang.Object l, java.lang.Object r )
```

  - **Usage**
    - \* Returns the value of an equal to operation
  - **Parameters**
    - \* **lc** - the class of the left operand
    - \* **rc** - the class of the right operand
    - \* **l** - the left operand
    - \* **r** - the right operand

---
- *getDeclaringClass*

```
public static Class getDeclaringClass( java.lang.Class c )
```

  - **Usage**
    - \* Returns the declaring class of the given class

---
- *getOuterField*

```
public static Field getOuterField( java.lang.Class cl, java.lang.String
                                   name )
```

  - **Usage**
    - \* Returns a field with the given name declared in one of the outer classes of the given class
  - **Parameters**
    - \* **cl** - the inner class
    - \* **name** - the name of the field

---
- *greaterOrEqual*

```
public static Object greaterOrEqual( java.lang.Object l, java.lang.Object
                                     r )
```

  - **Usage**
    - \* Returns the value of a greater or equal operation
  - **Parameters**
    - \* **l** - the left operand
    - \* **r** - the right operand

---
- *greaterThan*

```
public static Object greaterThan( java.lang.Object l, java.lang.Object r )
```

  - **Usage**

- \* Returns the value of a greater than operation
  - **Parameters**
    - \* **l** - the left operand
    - \* **r** - the right operand

---
- *isValidAssignment*

```
public static boolean isValidAssignment( java.lang.Class lc,
java.lang.Object val )
```

---
- *lessOrEqual*

```
public static Object lessOrEqual( java.lang.Object l, java.lang.Object r )
```

  - **Usage**
    - \* Returns the value of a less or equal operation
  - **Parameters**
    - \* **l** - the left operand
    - \* **r** - the right operand

---
- *lessThan*

```
public static Object lessThan( java.lang.Object l, java.lang.Object r )
```

  - **Usage**
    - \* Returns the value of a less than operation
  - **Parameters**
    - \* **l** - the left operand
    - \* **r** - the right operand

---
- *lookupOuterMethod*

```
public static Method lookupOuterMethod( java.lang.Class cl,
java.lang.String name, java.lang.Class [] ac )
```

  - **Usage**
    - \* Looks up for a method in an outer classes of this class.
  - **Parameters**
    - \* **cl** - the inner class
    - \* **name** - the name of the method
    - \* **ac** - the arguments classes (possibly not the exact declaring classes)

---
- *minus*

```
public static Object minus( java.lang.Class c, java.lang.Object o )
```

  - **Usage**
    - \* Returns the value of an unary - operation
  - **Parameters**
    - \* **c** - the class of the result
    - \* **o** - the operand

---
- *multiply*

```
public static Object multiply( java.lang.Class c, java.lang.Object l,
java.lang.Object r )
```

  - **Usage**

- \* Returns the value of a product
- **Parameters**
  - \* **c** - the class of the result
  - \* **l** - the left operand
  - \* **r** - the right operand

---
- *notEqualTo*

```
public static Object notEqualTo( java.lang.Class  lc, java.lang.Class  rc,
java.lang.Object  l, java.lang.Object  r )
```

  - **Usage**
    - \* Returns the value of a not equal to operation
  - **Parameters**
    - \* **lc** - the class of the left operand
    - \* **rc** - the class of the right operand
    - \* **l** - the left operand
    - \* **r** - the right operand

---
- *plus*

```
public static Object plus( java.lang.Class  c, java.lang.Object  o )
```

  - **Usage**
    - \* Returns the value of an unary + operation
  - **Parameters**
    - \* **c** - the class of the result
    - \* **o** - the operand

---
- *relationalOperation*

```
protected static Object relationalOperation( java.lang.Object  l,
java.lang.Object  r,
koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation  o )
```

  - **Usage**
    - \* Returns the value of a relational operation
  - **Parameters**
    - \* **l** - the left operand
    - \* **r** - the right operand
    - \* **o** - the operation

---
- *remainder*

```
public static Object remainder( java.lang.Class  c, java.lang.Object  l,
java.lang.Object  r )
```

  - **Usage**
    - \* Returns the value of remainder of a division
  - **Parameters**
    - \* **c** - the class of the result
    - \* **l** - the left operand
    - \* **r** - the right operand

---

- *shiftLeft*

```
public static Object shiftLeft( java.lang.Class  c, java.lang.Object  l,
java.lang.Object  r )
```

- **Usage**

- \* Returns the value of a shift left operation

- **Parameters**

- \* **c** - the class of the result
    - \* **l** - the left operand
    - \* **r** - the right operand

---

- *shiftOperation*

```
protected static Object shiftOperation( java.lang.Class  c, java.lang.Object
l, java.lang.Object  r,
koala.dynamicjava.interpreter.InterpreterUtilities.ShiftOperation  o )
```

- **Usage**

- \* Returns the value of a shift operation

- **Parameters**

- \* **c** - the class of the result
    - \* **l** - the left operand
    - \* **r** - the right operand
    - \* **o** - the operation

---

- *shiftRight*

```
public static Object shiftRight( java.lang.Class  c, java.lang.Object  l,
java.lang.Object  r )
```

- **Usage**

- \* Returns the value of a shift right operation

- **Parameters**

- \* **c** - the class of the result
    - \* **l** - the left operand
    - \* **r** - the right operand

---

- *subtract*

```
public static Object subtract( java.lang.Class  c, java.lang.Object  l,
java.lang.Object  r )
```

- **Usage**

- \* Returns the value of a subtraction

- **Parameters**

- \* **c** - the class of the result
    - \* **l** - the left operand
    - \* **r** - the right operand

---

- *unaryOperation*

```
public static Object unaryOperation( java.lang.Class  c, java.lang.Object
o, koala.dynamicjava.interpreter.InterpreterUtilities.UnaryOperation  u )
```

- **Usage**

- \* Returns the value of an unary operation

---

– **Parameters**

- \* **c** - the class of the result
  - \* **o** - the operand
  - \* **u** - the operation
- 

• *unsignedShiftRight*

```
public static Object unsignedShiftRight( java.lang.Class c,
java.lang.Object l, java.lang.Object r )
```

– **Usage**

- \* Returns the value of an unsigned shift right operation

– **Parameters**

- \* **c** - the class of the result
  - \* **l** - the left operand
  - \* **r** - the right operand
- 

• *xOr*

```
public static Object xOr( java.lang.Class c, java.lang.Object l,
java.lang.Object r )
```

– **Usage**

- \* Returns the value of a xor operation

– **Parameters**

- \* **c** - the class of the result
- \* **l** - the left operand
- \* **r** - the right operand

## 7.2.10 CLASS InterpreterUtilities.AddOperation

---

To encapsulate +

### DECLARATION

---

```
protected static class InterpreterUtilities.AddOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation
```

### CONSTRUCTORS

---

• *InterpreterUtilities.AddOperation*

```
protected InterpreterUtilities.AddOperation( )
```

### METHODS

---

• *invoke*

```
double invoke( double l, double r )
```

- *invoke*  
float invoke( float l, float r )
- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, long r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation

( in 7.2.11, page 462)

- *invoke*  
abstract double invoke( double l, double r )
- *invoke*  
abstract float invoke( float l, float r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

**7.2.11 CLASS InterpreterUtilities.BinaryArithmeticOperation**

To encapsulate a binary operator

## DECLARATION

```
protected abstract static class InterpreterUtilities.BinaryArithmeticOperation
extends java.lang.Object
```

## CONSTRUCTORS

- *InterpreterUtilities.BinaryArithmeticOperation*  
protected **InterpreterUtilities.BinaryArithmeticOperation**( )

## METHODS

- *invoke*  
abstract double invoke( double l, double r )
- *invoke*  
abstract float invoke( float l, float r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

### 7.2.12 CLASS InterpreterUtilities.BinaryPredicate

---

To encapsulate a boolean binary operator

#### DECLARATION

---

```
protected abstract static class InterpreterUtilities.BinaryPredicate
extends java.lang.Object
```

#### CONSTRUCTORS

---

- *InterpreterUtilities.BinaryPredicate*  
protected **InterpreterUtilities.BinaryPredicate**( )

#### METHODS

---

- *invoke*  
abstract boolean invoke( boolean l, boolean r )
- *invoke*  
abstract boolean invoke( double l, double r )
- *invoke*  
abstract boolean invoke( java.lang.Object l, java.lang.Object r )

### 7.2.13 CLASS InterpreterUtilities.BitAndOperation

---

To encapsulate &

#### DECLARATION

---

```
protected static class InterpreterUtilities.BitAndOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.BitwiseOperation
```

#### CONSTRUCTORS

---

- *InterpreterUtilities.BitAndOperation*  
protected **InterpreterUtilities.BitAndOperation**( )



METHODS

---

- *invoke*  
boolean invoke( boolean l, boolean r )
- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, long r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BitwiseOperation

---

( in 7.2.15, page 465)

- *invoke*  
abstract boolean invoke( boolean l, boolean r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

**7.2.14 CLASS InterpreterUtilities.BitOrOperation**

---

To encapsulate —

DECLARATION

---

```
protected static class InterpreterUtilities.BitOrOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.BitwiseOperation
```

CONSTRUCTORS

---

- *InterpreterUtilities.BitOrOperation*  
protected **InterpreterUtilities.BitOrOperation( )**

METHODS

---

- *invoke*  
boolean invoke( boolean l, boolean r )
- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, long r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BitwiseOperation

( in 7.2.15, page 465)

- *invoke*  
abstract boolean invoke( boolean l, boolean r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

**7.2.15 CLASS InterpreterUtilities.BitwiseOperation**

To encapsulate a bitwise operator

## DECLARATION

```
protected abstract static class InterpreterUtilities.BitwiseOperation
extends java.lang.Object
```

## CONSTRUCTORS

- *InterpreterUtilities.BitwiseOperation*  
protected **InterpreterUtilities.BitwiseOperation**( )

## METHODS

- *invoke*  
abstract boolean invoke( boolean l, boolean r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

**7.2.16 CLASS InterpreterUtilities.DivideOperation**

To encapsulate /

## DECLARATION

```
protected static class InterpreterUtilities.DivideOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation
```

CONSTRUCTORS

---

- *InterpreterUtilities.DivideOperation*  
protected **InterpreterUtilities.DivideOperation**( )

METHODS

---

- *invoke*  
double invoke( double l, double r )
- *invoke*  
float invoke( float l, float r )
- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, long r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation

---

( in 7.2.11, page 462)

- *invoke*  
abstract double invoke( double l, double r )
- *invoke*  
abstract float invoke( float l, float r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

**7.2.17 CLASS InterpreterUtilities.EqualToPredicate**

---

To encapsulate ==

DECLARATION

---

```
protected static class InterpreterUtilities.EqualToPredicate
extends koala.dynamicjava.interpreter.InterpreterUtilities.BinaryPredicate
```

CONSTRUCTORS

---

- *InterpreterUtilities.EqualToPredicate*  
protected **InterpreterUtilities.EqualToPredicate**( )

METHODS

---

- *invoke*  
boolean invoke( boolean l, boolean r )
- *invoke*  
boolean invoke( double l, double r )
- *invoke*  
boolean invoke( java.lang.Object l, java.lang.Object r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BinaryPredicate

---

( in 7.2.12, page 463)

- *invoke*  
abstract boolean invoke( boolean l, boolean r )
- *invoke*  
abstract boolean invoke( double l, double r )
- *invoke*  
abstract boolean invoke( java.lang.Object l, java.lang.Object r )

**7.2.18 CLASS InterpreterUtilities.GreaterOrEqualOperation**

---

To encapsulate &gt;=

DECLARATION

---

```
protected static class InterpreterUtilities.GreaterOrEqualOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation
```

CONSTRUCTORS

---

- *InterpreterUtilities.GreaterOrEqualOperation*  
protected **InterpreterUtilities.GreaterOrEqualOperation( )**

METHODS

---

- *invoke*  
boolean invoke( double l, double r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation

---

( in 7.2.26, page 473)

- *invoke*  
abstract boolean invoke( double l, double r )

### 7.2.19 CLASS *InterpreterUtilities.GreaterThanOperation*

---

To encapsulate >

#### DECLARATION

---

```
protected static class InterpreterUtilities.GreaterThanOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation
```

#### CONSTRUCTORS

---

- *InterpreterUtilities.GreaterThanOperation*  
protected **InterpreterUtilities.GreaterThanOperation**( )

#### METHODS

---

- *invoke*  
boolean **invoke**( double l, double r )

#### METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation

---

( in 7.2.26, page 473)

- *invoke*  
abstract boolean **invoke**( double l, double r )

### 7.2.20 CLASS *InterpreterUtilities.LessOrEqualOperation*

---

To encapsulate <=

#### DECLARATION

---

```
protected static class InterpreterUtilities.LessOrEqualOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation
```

#### CONSTRUCTORS

---

- *InterpreterUtilities.LessOrEqualOperation*  
protected **InterpreterUtilities.LessOrEqualOperation**( )

## METHODS

- *invoke*  
boolean **invoke**( double l, double r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation

( in 7.2.26, page 473)

- *invoke*  
abstract boolean **invoke**( double l, double r )

**7.2.21 CLASS InterpreterUtilities.LessThanOperation**

To encapsulate &lt;

## DECLARATION

```
protected static class InterpreterUtilities.LessThanOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation
```

## CONSTRUCTORS

- *InterpreterUtilities.LessThanOperation*  
protected **InterpreterUtilities.LessThanOperation**( )

## METHODS

- *invoke*  
boolean **invoke**( double l, double r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.RelationalOperation

( in 7.2.26, page 473)

- *invoke*  
abstract boolean **invoke**( double l, double r )

**7.2.22 CLASS InterpreterUtilities.MinusOperation**

To encapsulate -

## DECLARATION

---

```
protected static class InterpreterUtilities.MinusOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.UnaryOperation
```

---

## CONSTRUCTORS

- *InterpreterUtilities.MinusOperation*  
protected **InterpreterUtilities.MinusOperation**( )

## METHODS

- *invoke*  
double invoke( double o )
- *invoke*  
float invoke( float o )
- *invoke*  
int invoke( int o )
- *invoke*  
long invoke( long o )

## METHODS INHERITED FROM CLASS

```
koala.dynamicjava.interpreter.InterpreterUtilities.UnaryOperation
```

---

( in 7.2.32, page 477)

- *invoke*  
abstract double invoke( double o )
- *invoke*  
abstract float invoke( float o )
- *invoke*  
abstract int invoke( int o )
- *invoke*  
abstract long invoke( long o )

## 7.2.23 CLASS InterpreterUtilities.MultiplyOperation

---

To encapsulate \*

## DECLARATION

---

```
protected static class InterpreterUtilities.MultiplyOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation
```

---

CONSTRUCTORS

---

- *InterpreterUtilities.MultiplyOperation*  
protected **InterpreterUtilities.MultiplyOperation**( )

METHODS

---

- *invoke*  
double invoke( double l, double r )
- *invoke*  
float invoke( float l, float r )
- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, long r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation

---

( in 7.2.11, page 462)

- *invoke*  
abstract double invoke( double l, double r )
- *invoke*  
abstract float invoke( float l, float r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

**7.2.24 CLASS InterpreterUtilities.NotEqualToPredicate**

---

To encapsulate !=

DECLARATION

---

```
protected static class InterpreterUtilities.NotEqualToPredicate
extends koala.dynamicjava.interpreter.InterpreterUtilities.BinaryPredicate
```

CONSTRUCTORS

---

- *InterpreterUtilities.NotEqualToPredicate*  
protected **InterpreterUtilities.NotEqualToPredicate**( )



## METHODS

- *invoke*  
boolean invoke( boolean l, boolean r )
- *invoke*  
boolean invoke( double l, double r )
- *invoke*  
boolean invoke( java.lang.Object l, java.lang.Object r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BinaryPredicate

( in 7.2.12, page 463)

- *invoke*  
abstract boolean invoke( boolean l, boolean r )
- *invoke*  
abstract boolean invoke( double l, double r )
- *invoke*  
abstract boolean invoke( java.lang.Object l, java.lang.Object r )

## 7.2.25 CLASS InterpreterUtilities.PlusOperation

To encapsulate +

## DECLARATION

```
protected static class InterpreterUtilities.PlusOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.UnaryOperation
```

## CONSTRUCTORS

- *InterpreterUtilities.PlusOperation*  
protected **InterpreterUtilities.PlusOperation( )**

## METHODS

- *invoke*  
double invoke( double o )
- *invoke*  
float invoke( float o )
- *invoke*  
int invoke( int o )
- *invoke*  
long invoke( long o )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.UnaryOperation

( in 7.2.32, page 477)

- *invoke*  
abstract double invoke( double o )
- *invoke*  
abstract float invoke( float o )
- *invoke*  
abstract int invoke( int o )
- *invoke*  
abstract long invoke( long o )

**7.2.26 CLASS InterpreterUtilities.RelationalOperation**

To encapsulate a relational operation

## DECLARATION

```
protected abstract static class InterpreterUtilities.RelationalOperation
extends java.lang.Object
```

## CONSTRUCTORS

- *InterpreterUtilities.RelationalOperation*  
**protected InterpreterUtilities.RelationalOperation( )**

## METHODS

- *invoke*  
**abstract boolean invoke( double l, double r )**

**7.2.27 CLASS InterpreterUtilities.RemainderOperation**

To encapsulate %

## DECLARATION

```
protected static class InterpreterUtilities.RemainderOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation
```

CONSTRUCTORS

---

- *InterpreterUtilities.RemainderOperation*  
protected **InterpreterUtilities.RemainderOperation**( )

METHODS

---

- *invoke*  
double invoke( double l, double r )
- *invoke*  
float invoke( float l, float r )
- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, long r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation

---

( in 7.2.11, page 462)

- *invoke*  
abstract double invoke( double l, double r )
- *invoke*  
abstract float invoke( float l, float r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

**7.2.28 CLASS InterpreterUtilities.ShiftLeftOperation**

---

To encapsulate &lt;&lt;

DECLARATION

---

```
protected static class InterpreterUtilities.ShiftLeftOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.ShiftOperation
```

CONSTRUCTORS

---

- *InterpreterUtilities.ShiftLeftOperation*  
protected **InterpreterUtilities.ShiftLeftOperation**( )

## METHODS

- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, int r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.ShiftOperation

( in 7.2.29, page 475)

- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, int r )

**7.2.29 CLASS InterpreterUtilities.ShiftOperation**

To encapsulate a shift operator

## DECLARATION

```
protected abstract static class InterpreterUtilities.ShiftOperation
extends java.lang.Object
```

## CONSTRUCTORS

- *InterpreterUtilities.ShiftOperation*  
protected **InterpreterUtilities.ShiftOperation( )**

## METHODS

- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, int r )

**7.2.30 CLASS InterpreterUtilities.ShiftRightOperation**

To encapsulate &gt;&gt;

DECLARATION

---

```
protected static class InterpreterUtilities.ShiftRightOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.ShiftOperation
```

CONSTRUCTORS

---

- *InterpreterUtilities.ShiftRightOperation*  
protected **InterpreterUtilities.ShiftRightOperation**( )

METHODS

---

- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, int r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.ShiftOperation

---

( in 7.2.29, page 475)

- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, int r )

**7.2.31 CLASS InterpreterUtilities.SubtractOperation**

---

To encapsulate -

DECLARATION

---

```
protected static class InterpreterUtilities.SubtractOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation
```

CONSTRUCTORS

---

- *InterpreterUtilities.SubtractOperation*  
protected **InterpreterUtilities.SubtractOperation**( )

METHODS

---

- *invoke*  
double invoke( double l, double r )
- *invoke*  
float invoke( float l, float r )
- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, long r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BinaryArithmeticOperation

---

( in 7.2.11, page 462)

- *invoke*  
abstract double invoke( double l, double r )
- *invoke*  
abstract float invoke( float l, float r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, long r )

**7.2.32 CLASS InterpreterUtilities.UnaryOperation**

---

To encapsulate an unary operator

DECLARATION

---

```
protected abstract static class InterpreterUtilities.UnaryOperation
extends java.lang.Object
```

CONSTRUCTORS

---

- *InterpreterUtilities.UnaryOperation*  
protected **InterpreterUtilities.UnaryOperation( )**

METHODS

---

- *invoke*  
abstract double invoke( double o )
- *invoke*  
abstract float invoke( float o )
- *invoke*  
abstract int invoke( int o )
- *invoke*  
abstract long invoke( long o )

**7.2.33 CLASS InterpreterUtilities.UnsignedShiftRightOperation**

---

To encapsulate >>>

DECLARATION

---

protected static class InterpreterUtilities.UnsignedShiftRightOperation <b>extends</b> koala.dynamicjava.interpreter.InterpreterUtilities.ShiftOperation
---

CONSTRUCTORS

---

- *InterpreterUtilities.UnsignedShiftRightOperation*  
protected **InterpreterUtilities.UnsignedShiftRightOperation( )**

METHODS

---

- *invoke*  
int invoke( int l, int r )
- *invoke*  
long invoke( long l, int r )

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.ShiftOperation

---

( in 7.2.29, page 475)

- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long invoke( long l, int r )

### 7.2.34 CLASS InterpreterUtilities.XOrOperation

---

To encapsulate  $\wedge$

#### DECLARATION

---

```
protected static class InterpreterUtilities.XOrOperation
extends koala.dynamicjava.interpreter.InterpreterUtilities.BitwiseOperation
```

#### CONSTRUCTORS

---

- *InterpreterUtilities.XOrOperation*  
protected **InterpreterUtilities.XOrOperation**( )

#### METHODS

---

- *invoke*  
boolean invoke( boolean l, boolean r )
- *invoke*  
int invoke( int l, int r )
- *invoke*  
long **invoke**( long l, long r )

#### METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.InterpreterUtilities.BitwiseOperation

---

( in 7.2.15, page 465)

- *invoke*  
abstract boolean invoke( boolean l, boolean r )
- *invoke*  
abstract int invoke( int l, int r )
- *invoke*  
abstract long **invoke**( long l, long r )

### 7.2.35 CLASS Main

---

This file contains the entry point of the interpreter

#### DECLARATION

---

```
public class Main
extends java.lang.Object
```



## FIELDS

---

- private static String appname
  - The name of the application
- public static Interpreter interpreter
  -

## CONSTRUCTORS

---

- *Main*  
`public Main( )`

## METHODS

---

- *main*  
`public static void main( java.lang.String [] args )`
  - **Usage**
    - \* The main function
- *setClassPath*  
`private static void setClassPath( koala.dynamicjava.interpreter.Interpreter interp, java.lang.String cpath )`
  - **Usage**
    - \* Sets the class path for the given interpreter
- *setLibraryPath*  
`private static void setLibraryPath( koala.dynamicjava.interpreter.Interpreter interp, java.lang.String lpath )`
  - **Usage**
    - \* Sets the library path for the given interpreter
- *usage*  
`private static void usage( )`
  - **Usage**
    - \* Prints the usage

### 7.2.36 CLASS NameVisitor

---

This tree visitor resolves the ambiguity in identifiers in a syntax tree

DECLARATION

---

```
public class NameVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

FIELDS

---

- private Context context
  - The context

CONSTRUCTORS

---

- *NameVisitor*

```
public NameVisitor( koala.dynamicjava.interpreter.context.Context ctx )
```

  - **Usage**
    - \* Creates a new name visitor
  - **Parameters**
    - \* `ctx` - the context

METHODS

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

  - **Usage**
    - \* Visits an AddAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

  - **Usage**
    - \* Visits an AddExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

  - **Usage**
    - \* Visits an AndExpression
  - **Parameters**
    - \* `node` - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAccess node )  
 – **Usage**  
   \* Visits an ArrayAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAllocation node )  
 – **Usage**  
   \* Visits an ArrayAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayInitializer node )  
 – **Usage**  
   \* Visits a ArrayInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )  
 – **Usage**  
   \* Visits a BitAndAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )

- **Usage**
    - \* Visits a BitOrExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BlockStatement  node )
```

  - **Usage**
    - \* Visits a BlockStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.CastExpression  node )
```

  - **Usage**
    - \* Visits a CastExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.CatchStatement  node )
```

  - **Usage**
    - \* Visits a CatchStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassAllocation  node )
```

  - **Usage**
    - \* Visits a ClassAllocation
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration  node )
```

  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression  node )
```

  - **Usage**
    - \* Visits a ComplementExpression

- **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
```

    - **Usage**
      - \* Visits a ConditionalExpression
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
```

      - **Usage**
        - \* Visits an DivideAssignExpression
      - **Parameters**
        - \* `node` - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.DivideExpression node )
```

        - **Usage**
          - \* Visits a DivideExpression
        - **Parameters**
          - \* `node` - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.DoStatement node )
```

          - **Usage**
            - \* Visits a DoStatement
          - **Parameters**
            - \* `node` - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.EqualExpression node )
```

            - **Usage**
              - \* Visits an EqualExpression
            - **Parameters**
              - \* `node` - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
```

              - **Usage**
                - \* Visits a ExclusiveOrAssignExpression
              - **Parameters**

\* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
```

- **Usage**

- \* Visits a ExclusiveOrExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.FormalParameter node )
```

- **Usage**

- \* Visits a FormalParameter

- **Parameters**

- \* node - the node to visit

- **Returns** - the name of the parameter class

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ForStatement node )
```

- **Usage**

- \* Visits a ForStatement

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterExpression node )
```

- **Usage**

- \* Visits a GreaterExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )
```

- **Usage**

- \* Visits a GreaterOrEqualExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )
```

- **Usage**

- \* Visits an IfThenElseStatement

- **Parameters**

- \* node - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.IfThenStatement  node )
```

  - **Usage**
    - \* Visits an IfThenStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration  node )
```

  - **Usage**
    - \* Declares the package or class importation in the context
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.InnerAllocation  node )
```

  - **Usage**
    - \* Visits a InnerAllocation
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.InnerClassAllocation  node )
```

  - **Usage**
    - \* Visits an InnerClassAllocation
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression  node )
```

  - **Usage**
    - \* Visits a InstanceOfExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration  node )
```

  - **Usage**
    - \* Visits an InterfaceDeclaration
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.LabeledStatement  node )
```

- **Usage**
    - \* Visits a LabeledStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.LessExpression node )
```

  - **Usage**
    - \* Visits a LessExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
```

  - **Usage**
    - \* Visits a LessOrEqualExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MinusExpression node )
```

  - **Usage**
    - \* Visits a MinusExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

  - **Usage**
    - \* Visits an MultiplyAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

  - **Usage**
    - \* Visits a MultiplyExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

  - **Usage**
    - \* Visits a NotEqualExpression



- **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotExpression node )
```

    - **Usage**
      - \* Visits a NotExpression
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
```

      - **Usage**
        - \* Visits an ObjectFieldAccess
      - **Parameters**
        - \* `node` - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
```

        - **Usage**
          - \* Visits an ObjectMethodCall
        - **Parameters**
          - \* `node` - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.OrExpression node )
```

          - **Usage**
            - \* Visits an OrExpression
          - **Parameters**
            - \* `node` - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
```

            - **Usage**
              - \* Sets the context's current package
            - **Parameters**
              - \* `node` - the node to visit
            - **Returns** - null

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.PlusExpression node )
```

              - **Usage**
                - \* Visits a PlusExpression
              - **Parameters**

\* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.PostDecrement node )

- **Usage**

- \* Visits a PostDecrement

- **Parameters**

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.PostIncrement node )

- **Usage**

- \* Visits a PostIncrement

- **Parameters**

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.PreDecrement node )

- **Usage**

- \* Visits a PreDecrement

- **Parameters**

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.PreIncrement node )

- **Usage**

- \* Visits a PreIncrement

- **Parameters**

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.QualifiedName node )

- **Usage**

- \* Visits a QualifiedName

- **Parameters**

- \* node - the node to visit

- **Returns** - a node that depends of the meaning of this name. It could be : a QualifiedName, a ReferenceType or a FieldAccess.

---

- *visit*

public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )

- **Usage**

- \* Visits a RemainderAssignExpression

- **Parameters**

\* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderExpression node )
```

- **Usage**

- \* Visits a RemainderExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ReturnStatement node )
```

- **Usage**

- \* Visits a ReturnStatement

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
```

- **Usage**

- \* Visits a ShiftLeftAssignExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
```

- **Usage**

- \* Visits a ShiftLeftExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
```

- **Usage**

- \* Visits a ShiftRightAssignExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
```

- **Usage**

- \* Visits a ShiftRightExpression

- **Parameters**

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
```

- **Usage**

- \* Visits a SimpleAllocation

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )
```

- **Usage**

- \* Visits a SimpleAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
```

- **Usage**

- \* Visits an SubtractAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractExpression node )
```

- **Usage**

- \* Visits a SubtractExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperMethodCall node )
```

- **Usage**

- \* Visits a SuperMethodCall

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchBlock node )
```

- **Usage**

- \* Visits a SwitchBlock

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement node )
```

- **Usage**
    - \* Visits a SwitchStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement  node )
```

    - **Usage**
      - \* Visits a SynchronizedStatement
    - **Parameters**
      - \* **node** - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression  node )
```

      - **Usage**
        - \* Visits a ThisExpression
      - **Parameters**
        - \* **node** - the node to visit
      - **Returns** - a qualified name or a field access

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement  node )
```

        - **Usage**
          - \* Visits a ThrowStatement
        - **Parameters**
          - \* **node** - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement  node )
```

          - **Usage**
            - \* Visits a TryStatement
          - **Parameters**
            - \* **node** - the node to visit

---
        - *visit*

```
public Object visit(
koala.dynamicjava.tree.UnsignedShiftRightAssignExpression  node )
```

            - **Usage**
              - \* Visits a UnsignedShiftRightAssignExpression
            - **Parameters**
              - \* **node** - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression
node )
```

- **Usage**
    - \* Visits a UnsignedShiftRightExpression
  - **Parameters**
    - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration node )
```

    - **Usage**
      - \* Visits a VariableDeclaration
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.WhileStatement node )
```

    - **Usage**
      - \* Visits a WhileStatement
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visitBinaryExpression*

```
private void visitBinaryExpression( koala.dynamicjava.tree.BinaryExpression node )
```

    - **Usage**
      - \* Visits the subexpressions of a BinaryExpression
- 
- *visitExpressionContainer*

```
private void visitExpressionContainer( koala.dynamicjava.tree.ExpressionContainer node )
```

    - **Usage**
      - \* Visits an expression container
- 
- *visitList*

```
private void visitList( java.util.List l )
```

    - **Usage**
      - \* Visits a list of node

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.visitor.VisitorObject`

---

( in 5.2.1, page 160)

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

  - **Usage**
    - \* Visits an AddAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.AddExpression node )  
 – **Usage**  
   \* Visits a AddExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.AndExpression node )  
 – **Usage**  
   \* Visits a AndExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAccess node )  
 – **Usage**  
   \* Visits a ArrayAccess  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAllocation node )  
 – **Usage**  
   \* Visits an ArrayAllocation  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayInitializer node )  
 – **Usage**  
   \* Visits an ArrayInitializer  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayType node )  
 – **Usage**  
   \* Visits a ArrayType  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )  
 – **Usage**  
   \* Visits a BitAndAssignExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )

- **Usage**
    - \* Visits a BitAndExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )
```

  - **Usage**
    - \* Visits a BitOrAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrExpression node )
```

  - **Usage**
    - \* Visits a BitOrExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BlockStatement node )
```

  - **Usage**
    - \* Visits a BlockStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BreakStatement node )
```

  - **Usage**
    - \* Visits a BreakStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.CastExpression node )
```

  - **Usage**
    - \* Visits a CastExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.CatchStatement node )
```

  - **Usage**
    - \* Visits a CatchStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassAllocation node )
```

  - **Usage**
    - \* Visits an ClassAllocation
  - **Parameters**



- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassDeclaration node )  
 – **Usage**  
   \* Visits a ClassDeclaration  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassInitializer node )  
 – **Usage**  
   \* Visits a ClassInitializer  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ComplementExpression node )  
 – **Usage**  
   \* Visits a ComplementExpression  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConditionalExpression node )  
 – **Usage**  
   \* Visits a ConditionalExpression  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )  
 – **Usage**  
   \* Visits a ConstructorDeclaration  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )  
 – **Usage**  
   \* Visits a ConstructorInvocation  
 – **Parameters**  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.ContinueStatement node )  
 – **Usage**  
   \* Visits a ContinueStatement  
 – **Parameters**  
   \* node - the node to visit
-

- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )  
 – **Usage**  
   \* Visits an DivideAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideExpression node )  
 – **Usage**  
   \* Visits a DivideExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.DoStatement node )  
 – **Usage**  
   \* Visits a DoStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.EmptyStatement node )  
 – **Usage**  
   \* Visits an EmptyStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.EqualExpression node )  
 – **Usage**  
   \* Visits a EqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )  
 – **Usage**  
   \* Visits a ExclusiveOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )  
 – **Usage**  
   \* Visits a ExclusiveOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FieldDeclaration node )

- **Usage**
    - \* Visits a FieldDeclaration
  - **Parameters**
    - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.FormalParameter  node )
```

    - **Usage**
      - \* Visits a FormalParameter
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ForStatement  node )
```

    - **Usage**
      - \* Visits a ForStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.FunctionCall  node )
```

    - **Usage**
      - \* Visits a FunctionCall
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterExpression  node )
```

    - **Usage**
      - \* Visits a GreaterExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression  node )
```

    - **Usage**
      - \* Visits a GreaterOrEqualExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenElseStatement  node )
```

    - **Usage**
      - \* Visits a IfThenElseStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenStatement  node )
```

    - **Usage**
      - \* Visits a IfThenStatement
    - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.ImportDeclaration node )
      - Usage
        - \* Visits an ImportDeclaration
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.InnerAllocation node )
      - Usage
        - \* Visits an InnerAllocation
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )
      - Usage
        - \* Visits an InnerClassAllocation
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceInitializer node )
      - Usage
        - \* Visits a InstanceInitializer
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
      - Usage
        - \* Visits an InstanceOfExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )
      - Usage
        - \* Visits a InterfaceDeclaration
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.LabeledStatement node )
      - Usage
        - \* Visits a LabeledStatement
      - Parameters
        - \* node - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.LessExpression node )
```

  - **Usage**  
\* Visits a LessExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
```

  - **Usage**  
\* Visits a LessOrEqualExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.Literal node )
```

  - **Usage**  
\* Visits a Literal
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
```

  - **Usage**  
\* Visits a MethodDeclaration
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.MinusExpression node )
```

  - **Usage**  
\* Visits a MinusExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

  - **Usage**  
\* Visits an MultiplyAssignExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

  - **Usage**  
\* Visits a MultiplyExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

- **Usage**
    - \* Visits a NotEqualExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotExpression node )
```

  - **Usage**
    - \* Visits a NotExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
```

  - **Usage**
    - \* Visits a ObjectFieldAccess
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
```

  - **Usage**
    - \* Visits a ObjectMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.OrExpression node )
```

  - **Usage**
    - \* Visits a OrExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
```

  - **Usage**
    - \* Visits an PackageDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PlusExpression node )
```

  - **Usage**
    - \* Visits a PlusExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement node )
```

  - **Usage**
    - \* Visits a PostDecrement
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.PostIncrement node )
      - Usage
        - \* Visits a PostIncrement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.PreDecrement node )
      - Usage
        - \* Visits a PreDecrement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.PreIncrement node )
      - Usage
        - \* Visits a PreIncrement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.PrimitiveType node )
      - Usage
        - \* Visits a PrimitiveType
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.QualifiedName node )
      - Usage
        - \* Visits a QualifiedName
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ReferenceType node )
      - Usage
        - \* Visits a ReferenceType
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
      - Usage
        - \* Visits an RemainderAssignExpression
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderExpression node )  
 – **Usage**  
   \* Visits a RemainderExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ReturnStatement node )  
 – **Usage**  
   \* Visits a ReturnStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )  
 – **Usage**  
   \* Visits an ShiftLeftAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )  
 – **Usage**  
   \* Visits a ShiftLeftExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )  
 – **Usage**  
   \* Visits an ShiftRightAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )  
 – **Usage**  
   \* Visits a ShiftRightExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAllocation node )  
 – **Usage**  
   \* Visits an SimpleAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )  


---



```

    - Usage
      * Visits an SimpleAssignExpression
    - Parameters
      * node - the node to visit


---


    • visit
    public Object visit( koala.dynamicjava.tree.StaticFieldAccess  node )

    - Usage
      * Visits a StaticFieldAccess
    - Parameters
      * node - the node to visit


---


    • visit
    public Object visit( koala.dynamicjava.tree.StaticMethodCall  node )

    - Usage
      * Visits a StaticMethodCall
    - Parameters
      * node - the node to visit


---


    • visit
    public Object visit( koala.dynamicjava.tree.SubtractAssignExpression  node )

    - Usage
      * Visits an SubtractAssignExpression
    - Parameters
      * node - the node to visit


---


    • visit
    public Object visit( koala.dynamicjava.tree.SubtractExpression  node )

    - Usage
      * Visits a SubtractExpression
    - Parameters
      * node - the node to visit


---


    • visit
    public Object visit( koala.dynamicjava.tree.SuperFieldAccess  node )

    - Usage
      * Visits a SuperFieldAccess
    - Parameters
      * node - the node to visit


---


    • visit
    public Object visit( koala.dynamicjava.tree.SuperMethodCall  node )

    - Usage
      * Visits a SuperMethodCall
    - Parameters
      * node - the node to visit


---


    • visit
    public Object visit( koala.dynamicjava.tree.SwitchBlock  node )

    - Usage
      * Visits a SwitchBlock
    - Parameters

```

- 
- \* node - the node to visit
  - - *visit*
    - public Object visit( koala.dynamicjava.tree.SwitchStatement node )
    - Usage
      - \* Visits a SwitchStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )
    - Usage
      - \* Visits a SynchronizedStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ThisExpression node )
    - Usage
      - \* Visits a ThisExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ThrowStatement node )
    - Usage
      - \* Visits a ThrowStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.TryStatement node )
    - Usage
      - \* Visits a TryStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.TypeExpression node )
    - Usage
      - \* Visits a TypeExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )
    - Usage
      - \* Visits an UnsignedShiftRightAssignExpression
      - Parameters
      - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression node )
  - **Usage**  
 \* Visits a UnsignedShiftRightExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  
 public Object visit( koala.dynamicjava.tree.VariableDeclaration node )
  - **Usage**  
 \* Visits a VariableDeclaration
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  
 public Object visit( koala.dynamicjava.tree.WhileStatement node )
  - **Usage**  
 \* Visits a WhileStatement
  - **Parameters**  
 \* node - the node to visit

### 7.2.37 CLASS NodeProperties

---

This interface contains the names of the syntax tree properties defined by the interpretative kernel

#### DECLARATION

---

```
public class NodeProperties
extends java.lang.Object
```

#### FIELDS

---

- public static final String COMPONENT\_TYPE
  - The componentType property is defined for each array allocation. It contains a Class object.
- public static final String CONSTRUCTOR
  - The constructor property is defined for constructor call It contains a Constructor object
- public static final String FIELD
  - The field property is defined for field access node It contains a Field object
- public static final String METHOD
  - The method property is defined for method access node It contains a Method object
- public static final String FUNCTION

- The function property is defined for function access node It contains a MethodDeclaration object
- public static final String FUNCTIONS
  - The functions property is defined for function access node It contains a List object
- public static final String MODIFIER
  - The modifier property is defined for each variable. It contains a LeftHandSideModifier object.
- public static final String TYPE
  - The type property is defined for each expression It contains a Class object
- public static final String VARIABLES
  - The variables property is defined for each node where a new scope is entered. It contains a set of strings.
- public static final String VALUE
  - The value property is defined for each constant expression. It contains a java object.
- public static final String INNER\_ALLOCATION
  - The innerClass property
- public static final String OUTER\_INNER\_ALLOCATION
  - The outerInnerClass property
- public static final String IMPORTATION\_MANAGER
  - The importation manager property
- public static final String INSTANCE\_INITIALIZER
  - The instanceInitializer property
- public static final String ERROR\_STRINGS
  - The errorStrings property contains an array of additional messages

## CONSTRUCTORS

---

- *NodeProperties*  
**protected NodeProperties( )**
  - **Usage**
    - \* This class contains only static method and constants, so it is not useful to create instances of it.

## METHODS

- *getClassInfo*  
`public static ClassInfo getClassInfo( koala.dynamicjava.tree.Node n )`  
 – **Usage**  
 \* Returns the type property of a node when it is a class info

---

- *getComponentType*  
`public static Class getComponentType( koala.dynamicjava.tree.Node n )`  
 – **Usage**  
 \* Returns the type property of a node

---

- *getModifier*  
`public static LeftHandSideModifier getModifier( koala.dynamicjava.tree.Node n )`  
 – **Usage**  
 \* Returns the modifier property of a node

---

- *getType*  
`public static Class getType( koala.dynamicjava.tree.Node n )`  
 – **Usage**  
 \* Returns the type property of a node

## 7.2.38 CLASS TreeClassFinder

The instances of the classes that implements this interface are used to find the fully qualified name of classes and to manage the loading of these classes.

## DECLARATION

```
public class TreeClassFinder
extends java.lang.Object
implements koala.dynamicjava.classinfo.ClassFinder
```

## CONSTRUCTORS

- *TreeClassFinder*  
`public TreeClassFinder( koala.dynamicjava.interpreter.context.Context ctx,  
 koala.dynamicjava.interpreter.Interpreter i,  
 koala.dynamicjava.interpreter.ClassPool cp )`  
 – **Usage**  
 \* Creates a new class finder  
 – **Parameters**

\* **ctx** - the context  
 \* **i** - the current interpreter  
 \* **cp** - the class pool

## METHODS

---

- *addClassInfo*

```
public ClassInfo addClassInfo( java.lang.String  cname,
koala.dynamicjava.tree.TypeDeclaration  decl )
```

- **Usage**

\* Adds a type declaration in the class info list

- **Parameters**

\* **cname** - the name of the class  
 \* **decl** - the type declaration

---

- *getCurrentPackage*

```
public String getCurrentPackage( )
```

- **Usage**

\* Returns the current package

---

- *getImportationManager*

```
public ImportationManager getImportationManager( )
```

- **Usage**

\* Returns the importation manager

---

- *getInterpreter*

```
public Interpreter getInterpreter( )
```

- **Usage**

\* Returns the current interpreter

---

- *lookupClass*

```
public ClassInfo lookupClass( java.lang.String  cname )
```

- **Usage**

\* Loads the class info that match the given name in the source file

- **Parameters**

\* **cname** - the name of the class to find

- **Returns** - the class info

- **Exceptions**

\* `java.lang.ClassNotFoundException` - if the class cannot be loaded

---

- *lookupClass*

```
public ClassInfo lookupClass( java.lang.String  cname,
koala.dynamicjava.classinfo.ClassInfo  cinfo )
```

- **Usage**

\* Loads the class info that match the given name in the source file

- **Parameters**

- \* **cname** - the name of the class to find
- \* **cinfo** - the context where 'cname' was found

- **Returns** - the class info

- **Exceptions**

- \* `java.lang.ClassNotFoundException` - if the class cannot be loaded

## 7.2.39 CLASS TreeClassLoader

---

This class is responsible for loading bytecode classes

All classes *created* by **TreeClassLoaders** have an identical `CodeSource` . This code source has no certificates but may have a location. The latter is set to the value of the system property with the value of `#CODE_SOURCE_URL_PROPERTY` as key, if it is a valid URL. If the property is defined but its value is not a valid URL, the location of the code source is set to `null`. If the property is not defined, the value of `#DEFAULT_CODE_SOURCE_URL` is used as location.

### DECLARATION

---

```
public class TreeClassLoader
extends java.security.SecureClassLoader
implements ClassLoaderContainer
```

### FIELDS

---

- public static String `DEFAULT_CODE_SOURCE_URL`
  - The default value for the `java.security.CodeSource` URL. May be overridden by setting the system property with name given by the value of `#CODE_SOURCE_URL_PROPERTY` .
- public static String `CODE_SOURCE_URL_PROPERTY`
  - Name of the system property to define the value for the URL of the `java.security.CodeSource` assigned to classes created by this classloader. The default value is the value of `#DEFAULT_CODE_SOURCE_URL` .

If the property value is not a wellformed URL, the URL is set to `null`.

### CONSTRUCTORS

---

- *TreeClassLoader*  

```
public TreeClassLoader( koala.dynamicjava.interpreter.Interpreter i )
```

- **Usage**

- \* Creates a new class loader

- **Parameters**

- \* *i* - the object used to interpret the classes

---

- *TreeClassLoader*

```
public TreeClassLoader( koala.dynamicjava.interpreter.Interpreter i,
java.lang.ClassLoader cl )
```

- **Usage**

- \* Creates a new class loader

- **Parameters**

- \* *i* - the object used to interpret the classes

- \* *cl* - the auxiliary class loader used to load external classes

## METHODS

---

- *addTree*

```
public void addTree( java.lang.String name,
koala.dynamicjava.tree.TypeDeclaration node )
```

- **Usage**

- \* Adds a class syntax tree to the list of the loaded trees

- **Parameters**

- \* *name* - the name of the type

- \* *node* - the tree

---

- *addURL*

```
public void addURL( java.net.URL url )
```

- **Usage**

- \* Adds an URL in the class path

---

- *defineClass*

```
public Class defineClass( java.lang.String name, byte [] code )
```

- **Usage**

- \* Converts an array of bytes into an instance of class Class and links this class.

- **Exceptions**

- \* *java.lang.ClassFormatError* - if the class could not be defined

---

- *findClass*

```
protected Class findClass( java.lang.String name )
```

- **Usage**

- \* Finds the specified class.

- **Parameters**

- \* *name* - the name of the class

- **Returns** - the resulting Class object

- **Exceptions**



\* `java.lang.ClassNotFoundException` - if the class could not be find

---

- *getClassLoader*

`public ClassLoader getClassLoader( )`

- **Usage**

\* Returns the additional class loader that is used for loading classes from the net.

- **Returns** - null if there is no additional class loader

---

- *getClassNames*

`public Set getClassNames( )`

- **Usage**

\* Returns the names of the defined classes in a set

---

- *getTree*

`public TypeDeclaration getTree( java.lang.String name )`

- **Usage**

\* Gets a tree

---

- *hasDefined*

`public boolean hasDefined( java.lang.String name )`

- **Usage**

\* Whether a class was defined by this class loader

#### METHODS INHERITED FROM CLASS `java.security.SecureClassLoader`

---

- *<clinit>*

`static void <clinit>( )`

- *check*

`private void check( )`

- *defineClass*

`protected final Class defineClass( java.lang.String , byte [] , int , int ,  
java.security.CodeSource )`

- *getPermissions*

`protected PermissionCollection getPermissions( java.security.CodeSource )`

- *getProtectionDomain*

`private ProtectionDomain getProtectionDomain( java.security.CodeSource )`

#### METHODS INHERITED FROM CLASS `java.lang.ClassLoader`

---

- *<clinit>*

`static void <clinit>( )`

- *addClass*

`void addClass( java.lang.Class )`

- *check*

`private void check( )`

---

- *checkCerts*  
private synchronized void checkCerts( java.lang.String , java.security.CodeSource )
- *checkPackageAccess*  
private void checkPackageAccess( java.lang.Class , java.security.ProtectionDomain )
- *clearAssertionStatus*  
public synchronized void clearAssertionStatus( )
- *compareCerts*  
private boolean compareCerts( java.security.cert.Certificate [] , java.security.cert.Certificate [] )
- *defineClass*  
protected final Class defineClass( byte [] , int , int )
- *defineClass*  
protected final Class defineClass( java.lang.String , byte [] , int , int )
- *defineClass*  
protected final Class defineClass( java.lang.String , byte [] , int , int , java.security.ProtectionDomain )
- *defineClass0*  
private native Class defineClass0( java.lang.String , byte [] , int , int , java.security.ProtectionDomain )
- *definePackage*  
protected Package definePackage( java.lang.String , java.lang.String , java.lang.String , java.lang.String , java.lang.String , java.lang.String , java.net.URL )
- *desiredAssertionStatus*  
synchronized boolean desiredAssertionStatus( java.lang.String )
- *findBootstrapClass*  
private native Class findBootstrapClass( java.lang.String )
- *findBootstrapClass0*  
private Class findBootstrapClass0( java.lang.String )
- *findClass*  
protected Class findClass( java.lang.String )
- *findLibrary*  
protected String findLibrary( java.lang.String )
- *findLoadedClass*  
protected final native Class findLoadedClass( java.lang.String )
- *findNative*  
static long findNative( java.lang.ClassLoader , java.lang.String )
- *findResource*  
protected URL findResource( java.lang.String )
- *findResources*  
protected Enumeration findResources( java.lang.String )
- *findSystemClass*  
protected final Class findSystemClass( java.lang.String )
- *getBootstrapClassPath*  
static URLClassPath getBootstrapClassPath( )
- *getBootstrapResource*  
private static URL getBootstrapResource( java.lang.String )
- *getBootstrapResources*  
private static Enumeration getBootstrapResources( java.lang.String )

- *getCallerClassLoader*  
static ClassLoader getCallerClassLoader( )
- *getDefaultDomain*  
private synchronized ProtectionDomain getDefaultDomain( )
- *getPackage*  
protected Package getPackage( java.lang.String )
- *getPackages*  
protected Package getPackages( )
- *getParent*  
public final ClassLoader getParent( )
- *getResource*  
public URL getResource( java.lang.String )
- *getResourceAsStream*  
public InputStream getResourceAsStream( java.lang.String )
- *getResources*  
public final Enumeration getResources( java.lang.String )
- *getSystemClassLoader*  
public static ClassLoader getSystemClassLoader( )
- *getSystemResource*  
public static URL getSystemResource( java.lang.String )
- *getSystemResourceAsStream*  
public static InputStream getSystemResourceAsStream( java.lang.String )
- *getSystemResources*  
public static Enumeration getSystemResources( java.lang.String )
- *initializeJavaAssertionMaps*  
private void initializeJavaAssertionMaps( )
- *initializePath*  
private static String initializePath( java.lang.String )
- *initSystemClassLoader*  
private static synchronized void initSystemClassLoader( )
- *isAncestor*  
boolean isAncestor( java.lang.ClassLoader )
- *loadClass*  
public Class loadClass( java.lang.String )
- *loadClass*  
protected synchronized Class loadClass( java.lang.String , boolean )
- *loadClassInternal*  
private synchronized Class loadClassInternal( java.lang.String )
- *loadLibrary*  
static void loadLibrary( java.lang.Class , java.lang.String , boolean )
- *loadLibrary0*  
private static boolean loadLibrary0( java.lang.Class , java.io.File )
- *registerNatives*  
private static native void registerNatives( )
- *resolveClass*  
protected final void resolveClass( java.lang.Class )
- *resolveClass0*  
private native void resolveClass0( java.lang.Class )
- *retrieveDirectives*  
private static native AssertionStatusDirectives retrieveDirectives( )

- *setClassAssertionStatus*  
public synchronized void setClassAssertionStatus( java.lang.String , boolean )
- *setDefaultAssertionStatus*  
public synchronized void setDefaultAssertionStatus( boolean )
- *setPackageAssertionStatus*  
public synchronized void setPackageAssertionStatus( java.lang.String , boolean )
- *setSigners*  
protected final void setSigners( java.lang.Class , java.lang.Object [] )

## 7.2.40 CLASS TreeCompiler

---

This class contains methods to manage the creation of classes.

### DECLARATION

---

```
public class TreeCompiler
extends java.lang.Object
```

### CONSTRUCTORS

---

- *TreeCompiler*  
public **TreeCompiler**( koala.dynamicjava.interpreter.Interpreter i )  
  - **Usage**  
\* Creates a new compiler
  - **Parameters**  
\* i - the current interpreter

### METHODS

---

- *classExists*  
protected boolean **classExists**( java.lang.String name )  
  - **Usage**  
\* Whether a class exists in a compiled form
- *compile*  
public Class **compile**( java.lang.String name )  
  - **Usage**  
\* Compiles a compilation unit
  - **Parameters**  
\* name - the name of the class to compile
- *compileClass*  
protected Class **compileClass**( koala.dynamicjava.classinfo.ClassInfo ci,  
java.lang.String name )

- **Usage**
  - \* Compiles the given class info
- **Parameters**
  - \* **ci** - the class info to compile
  - \* **name** - the name of the class to return

---

- *compileClasses*

```
public Class compileClasses( java.lang.String name )
```

- **Usage**
  - \* Compiles all the classes in the class pool
- **Parameters**
  - \* **name** - the name of the class to return

---

- *compileTree*

```
public Class compileTree( koala.dynamicjava.interpreter.context.Context
ctx, koala.dynamicjava.tree.TypeDeclaration td )
```

- **Usage**
  - \* Compiles a single class
- **Parameters**
  - \* **td** - the type declaration
  - \* **im** - the importation manager

---

- *loadClass*

```
protected void loadClass( java.lang.String name )
```

- **Usage**
  - \* Searches for a class, loads its class info structure

## 7.2.41 CLASS TreeCompiler.ClassInfoLoader

---

To load class infos instead of classes

### DECLARATION

---

```
protected class TreeCompiler.ClassInfoLoader
extends java.lang.ClassLoader
```

### CONSTRUCTORS

---

- *TreeCompiler.ClassInfoLoader*  

```
protected TreeCompiler.ClassInfoLoader( )
```

METHODS

---

- *findClass*  
`protected Class findClass( java.lang.String name )`
  - **Usage**
    - \* Finds the specified class.
  - **Parameters**
    - \* `name` - the name of the class
  - **Returns** - the resulting `Class` object
  - **Exceptions**
    - \* `java.lang.ClassNotFoundException` - if the class could not be find

METHODS INHERITED FROM CLASS `java.lang.ClassLoader`

---

- *<clinit>*  
`static void <clinit>( )`
- *addClass*  
`void addClass( java.lang.Class )`
- *check*  
`private void check( )`
- *checkCerts*  
`private synchronized void checkCerts( java.lang.String , java.security.CodeSource )`
- *checkPackageAccess*  
`private void checkPackageAccess( java.lang.Class , java.security.ProtectionDomain )`
- *clearAssertionStatus*  
`public synchronized void clearAssertionStatus( )`
- *compareCerts*  
`private boolean compareCerts( java.security.cert.Certificate [] , java.security.cert.Certificate [] )`
- *defineClass*  
`protected final Class defineClass( byte [] , int , int )`
- *defineClass*  
`protected final Class defineClass( java.lang.String , byte [] , int , int )`
- *defineClass*  
`protected final Class defineClass( java.lang.String , byte [] , int , int , java.security.ProtectionDomain )`
- *defineClass0*  
`private native Class defineClass0( java.lang.String , byte [] , int , int , java.security.ProtectionDomain )`
- *definePackage*  
`protected Package definePackage( java.lang.String , java.lang.String , java.lang.String , java.lang.String , java.lang.String , java.net.URL )`
- *desiredAssertionStatus*  
`synchronized boolean desiredAssertionStatus( java.lang.String )`

- *findBootstrapClass*  
private native Class findBootstrapClass( java.lang.String    )
- *findBootstrapClass0*  
private Class findBootstrapClass0( java.lang.String    )
- *findClass*  
protected Class findClass( java.lang.String    )
- *findLibrary*  
protected String findLibrary( java.lang.String    )
- *findLoadedClass*  
protected final native Class findLoadedClass( java.lang.String    )
- *findNative*  
static long findNative( java.lang.ClassLoader    , java.lang.String    )
- *findResource*  
protected URL findResource( java.lang.String    )
- *findResources*  
protected Enumeration findResources( java.lang.String    )
- *findSystemClass*  
protected final Class findSystemClass( java.lang.String    )
- *getBootstrapClassPath*  
static URLClassPath getBootstrapClassPath(    )
- *getBootstrapResource*  
private static URL getBootstrapResource( java.lang.String    )
- *getBootstrapResources*  
private static Enumeration getBootstrapResources( java.lang.String    )
- *getCallerClassLoader*  
static ClassLoader getCallerClassLoader(    )
- *getDefaultDomain*  
private synchronized ProtectionDomain getDefaultDomain(    )
- *getPackage*  
protected Package getPackage( java.lang.String    )
- *getPackages*  
protected Package getPackages(    )
- *getParent*  
public final ClassLoader getParent(    )
- *getResource*  
public URL getResource( java.lang.String    )
- *getResourceAsStream*  
public InputStream getResourceAsStream( java.lang.String    )
- *getResources*  
public final Enumeration getResources( java.lang.String    )
- *getSystemClassLoader*  
public static ClassLoader getSystemClassLoader(    )
- *getSystemResource*  
public static URL getSystemResource( java.lang.String    )
- *getSystemResourceAsStream*  
public static InputStream getSystemResourceAsStream( java.lang.String    )
- *getSystemResources*  
public static Enumeration getSystemResources( java.lang.String    )
- *initializeJavaAssertionMaps*  
private void initializeJavaAssertionMaps(    )

- *initializePath*  
private static String initializePath( java.lang.String )
- *initSystemClassLoader*  
private static synchronized void initSystemClassLoader( )
- *isAncestor*  
boolean isAncestor( java.lang.ClassLoader )
- *loadClass*  
public Class loadClass( java.lang.String )
- *loadClass*  
protected synchronized Class loadClass( java.lang.String , boolean )
- *loadClassInternal*  
private synchronized Class loadClassInternal( java.lang.String )
- *loadLibrary*  
static void loadLibrary( java.lang.Class , java.lang.String , boolean )
- *loadLibrary0*  
private static boolean loadLibrary0( java.lang.Class , java.io.File )
- *registerNatives*  
private static native void registerNatives( )
- *resolveClass*  
protected final void resolveClass( java.lang.Class )
- *resolveClass0*  
private native void resolveClass0( java.lang.Class )
- *retrieveDirectives*  
private static native AssertionStatusDirectives retrieveDirectives( )
- *setClassAssertionStatus*  
public synchronized void setClassAssertionStatus( java.lang.String , boolean )
- *setDefaultAssertionStatus*  
public synchronized void setDefaultAssertionStatus( boolean )
- *setPackageAssertionStatus*  
public synchronized void setPackageAssertionStatus( java.lang.String , boolean )
- *setSigners*  
protected final void setSigners( java.lang.Class , java.lang.Object [] )

## 7.2.42 CLASS TreeCompiler.CompilationUnitVisitor

---

To create the class infos for a compilation unit

### DECLARATION

---

```
protected class TreeCompiler.CompilationUnitVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

### CONSTRUCTORS

---

- *TreeCompiler.CompilationUnitVisitor*  
protected TreeCompiler.CompilationUnitVisitor( )



METHODS

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration  node )
```

– **Usage**

\* Visits a ClassDeclaration

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration  node )
```

– **Usage**

\* Visits an ImportDeclaration

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration  node )
```

– **Usage**

\* Visits an InterfaceDeclaration

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration  node )
```

– **Usage**

\* Visits a PackageDeclaration

– **Parameters**

\* **node** - the node to visit

– **Returns** - null• *visitType*

```
protected Object visitType( koala.dynamicjava.tree.TypeDeclaration  node )
```

– **Usage**

\* Visits a type declaration

– **Parameters**

\* **node** - the node to visit

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.visitor.VisitorObject`

---

( in 5.2.1, page 160)

- *visit*  
`public Object visit( koala.dynamicjava.tree.AddAssignExpression node )`  
  - **Usage**  
\* Visits an AddAssignExpression
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.AddExpression node )`  
  - **Usage**  
\* Visits a AddExpression
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.AndExpression node )`  
  - **Usage**  
\* Visits a AndExpression
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayAccess node )`  
  - **Usage**  
\* Visits a ArrayAccess
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayAllocation node )`  
  - **Usage**  
\* Visits an ArrayAllocation
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayInitializer node )`  
  - **Usage**  
\* Visits an ArrayInitializer
  - **Parameters**  
\* `node` - the node to visit

---
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayType node )`  
  - **Usage**  
\* Visits a ArrayType
  - **Parameters**  
\* `node` - the node to visit

- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )  
 – **Usage**  
   \* Visits a BitAndAssignExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )  
 – **Usage**  
   \* Visits a BitOrExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )  
 – **Usage**  
   \* Visits a BlockStatement  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.BreakStatement node )  
 – **Usage**  
   \* Visits a BreakStatement  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.CastExpression node )  
 – **Usage**  
   \* Visits a CastExpression  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object visit( koala.dynamicjava.tree.CatchStatement node )

- **Usage**
    - \* Visits a CatchStatement
  - **Parameters**
    - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassAllocation node )
```

    - **Usage**
      - \* Visits an ClassAllocation
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration node )
```

    - **Usage**
      - \* Visits a ClassDeclaration
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer node )
```

    - **Usage**
      - \* Visits a ClassInitializer
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression node )
```

    - **Usage**
      - \* Visits a ComplementExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
```

    - **Usage**
      - \* Visits a ConditionalExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )
```

    - **Usage**
      - \* Visits a ConstructorDeclaration
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )
```

    - **Usage**
      - \* Visits a ConstructorInvocation
    - **Parameters**

- 
- \* node - the node to visit

---

• *visit*

public Object visit( koala.dynamicjava.tree.ContinueStatement node )

    - Usage
    - \* Visits a ContinueStatement
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )

    - Usage
    - \* Visits an DivideAssignExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.DivideExpression node )

    - Usage
    - \* Visits a DivideExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.DoStatement node )

    - Usage
    - \* Visits a DoStatement
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.EmptyStatement node )

    - Usage
    - \* Visits an EmptyStatement
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.EqualExpression node )

    - Usage
    - \* Visits a EqualExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )

    - Usage
    - \* Visits a ExclusiveOrAssignExpression
    - Parameters
    - \* node - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
```

  - **Usage**  
\* Visits a ExclusiveOrExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.FieldDeclaration node )
```

  - **Usage**  
\* Visits a FieldDeclaration
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.FormalParameter node )
```

  - **Usage**  
\* Visits a FormalParameter
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ForStatement node )
```

  - **Usage**  
\* Visits a ForStatement
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.FunctionCall node )
```

  - **Usage**  
\* Visits a FunctionCall
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.GreaterExpression node )
```

  - **Usage**  
\* Visits a GreaterExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )
```

  - **Usage**  
\* Visits a GreaterOrEqualExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )
```

- **Usage**
    - \* Visits a IfThenElseStatement
  - **Parameters**
    - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenStatement  node )
```

    - **Usage**
      - \* Visits a IfThenStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration  node )
```

    - **Usage**
      - \* Visits an ImportDeclaration
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerAllocation  node )
```

    - **Usage**
      - \* Visits an InnerAllocation
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerClassAllocation  node )
```

    - **Usage**
      - \* Visits an InnerClassAllocation
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceInitializer  node )
```

    - **Usage**
      - \* Visits a InstanceInitializer
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression  node )
```

    - **Usage**
      - \* Visits an InstanceOfExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration  node )
```

    - **Usage**
      - \* Visits a InterfaceDeclaration
    - **Parameters**

- 
- \* node - the node to visit

---

• *visit*

public Object visit( koala.dynamicjava.tree.LabeledStatement node )

    - Usage
    - \* Visits a LabeledStatement
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.LessExpression node )

    - Usage
    - \* Visits a LessExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )

    - Usage
    - \* Visits a LessOrEqualExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.Literal node )

    - Usage
    - \* Visits a Literal
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.MethodDeclaration node )

    - Usage
    - \* Visits a MethodDeclaration
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.MinusExpression node )

    - Usage
    - \* Visits a MinusExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )

    - Usage
    - \* Visits an MultiplyAssignExpression
    - Parameters
    - \* node - the node to visit

---



- *visit*  

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

  - **Usage**  
\* Visits a MultiplyExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

  - **Usage**  
\* Visits a NotEqualExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.NotExpression node )
```

  - **Usage**  
\* Visits a NotExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
```

  - **Usage**  
\* Visits a ObjectFieldAccess
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
```

  - **Usage**  
\* Visits a ObjectMethodCall
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.OrExpression node )
```

  - **Usage**  
\* Visits a OrExpression
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
```

  - **Usage**  
\* Visits an PackageDeclaration
  - **Parameters**  
\* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.PlusExpression node )
```

- **Usage**
    - \* Visits a PlusExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement node )
```

  - **Usage**
    - \* Visits a PostDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement node )
```

  - **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.QualifiedName node )
```

  - **Usage**
    - \* Visits a QualifiedName
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReferenceType node )
```

  - **Usage**
    - \* Visits a ReferenceType
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*
    - public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
    - Usage
      - \* Visits an RemainderAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.RemainderExpression node )
    - Usage
      - \* Visits a RemainderExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ReturnStatement node )
    - Usage
      - \* Visits a ReturnStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
    - Usage
      - \* Visits an ShiftLeftAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
    - Usage
      - \* Visits a ShiftLeftExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
    - Usage
      - \* Visits an ShiftRightAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
    - Usage
      - \* Visits a ShiftRightExpression
      - Parameters
      - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAllocation node )  
 – **Usage**  
   \* Visits an SimpleAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )  
 – **Usage**  
   \* Visits an SimpleAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )  
 – **Usage**  
   \* Visits a StaticFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticMethodCall node )  
 – **Usage**  
   \* Visits a StaticMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )  
 – **Usage**  
   \* Visits an SubtractAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractExpression node )  
 – **Usage**  
   \* Visits a SubtractExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )  
 – **Usage**  
   \* Visits a SuperFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperMethodCall node )  


---

- **Usage**
    - \* Visits a SuperMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchBlock  node )
```

  - **Usage**
    - \* Visits a SwitchBlock
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement  node )
```

  - **Usage**
    - \* Visits a SwitchStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement  node )
```

  - **Usage**
    - \* Visits a SynchronizedStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression  node )
```

  - **Usage**
    - \* Visits a ThisExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement  node )
```

  - **Usage**
    - \* Visits a ThrowStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement  node )
```

  - **Usage**
    - \* Visits a TryStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression  node )
```

  - **Usage**
    - \* Visits a TypeExpression
  - **Parameters**

- 
- \* **node** - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )  
 )
    - **Usage**
      - \* Visits an UnsignedShiftRightAssignExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression node )  
 )
    - **Usage**
      - \* Visits a UnsignedShiftRightExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.VariableDeclaration node )  
 )
    - **Usage**
      - \* Visits a VariableDeclaration
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.WhileStatement node )  
 )
    - **Usage**
      - \* Visits a WhileStatement
    - **Parameters**
      - \* **node** - the node to visit

### 7.2.43 CLASS TreeCompiler.PseudoError

---

To test the existence of a class without loading it

#### DECLARATION

---

```
public class TreeCompiler.PseudoError
extends java.lang.Error
```

#### CONSTRUCTORS

---

- *TreeCompiler.PseudoError*  
 TreeCompiler.PseudoError( koala.dynamicjava.classinfo.ClassInfo ci )  
 )
  - **Usage**
    - \* Creates a new error

METHODS

---

- *getClassInfo*  
 public ClassInfo getClassInfo( )  
 – Usage  
 \* Returns the class info

METHODS INHERITED FROM CLASS java.lang.Error

---

METHODS INHERITED FROM CLASS java.lang.Throwable

---

- *fillInStackTrace*  
 public synchronized native Throwable fillInStackTrace( )
- *getCause*  
 public Throwable getCause( )
- *getLocalizedMessage*  
 public String getLocalizedMessage( )
- *getMessage*  
 public String getMessage( )
- *getOurStackTrace*  
 private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
 public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
 private native int getStackTraceDepth( )
- *getStackTraceElement*  
 private native StackTraceElement getStackTraceElement( int )
- *initCause*  
 public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
 public void printStackTrace( )
- *printStackTrace*  
 public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
 public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
 private void printStackTraceAsCause( java.io.PrintStream ,  
 java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
 private void printStackTraceAsCause( java.io.PrintWriter ,  
 java.lang.StackTraceElement [] )
- *setStackTrace*  
 public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
 public String toString( )
- *writeObject*  
 private synchronized void writeObject( java.io.ObjectOutputStream )

## 7.2.44 CLASS *TreeInterpreter*

---

This class contains method to interpret the constructs of the language.

### DECLARATION

---

```
public class TreeInterpreter
extends java.lang.Object
implements Interpreter
```

### CONSTRUCTORS

---

- *TreeInterpreter*  
public **TreeInterpreter**( koala.dynamicjava.parser.wrapper.ParserFactory pf )
  - **Usage**
    - \* Creates a new interpreter
  - **Parameters**
    - \* pf - the parser factory

---
- *TreeInterpreter*  
public **TreeInterpreter**( koala.dynamicjava.parser.wrapper.ParserFactory pf,  
java.lang.ClassLoader cl )
  - **Usage**
    - \* Creates a new interpreter
  - **Parameters**
    - \* pf - the parser factory
    - \* cl - the auxiliary class loader used to load external classes

### METHODS

---

- *addClassPath*  
public void **addClassPath**( java.lang.String path )
  - **Usage**
    - \* Adds a class search path
  - **Parameters**
    - \* path - the path to add

---
- *addClassURL*  
public void **addClassURL**( java.net.URL url )
  - **Usage**
    - \* Adds a class search URL
  - **Parameters**



---

\* url - the url to add

---

- *addLibraryPath*

public void **addLibraryPath**( java.lang.String path )

- **Usage**

- \* Adds a library search path

- **Parameters**

- \* path - the path to add

---

- *addLibrarySuffix*

public void **addLibrarySuffix**( java.lang.String s )

- **Usage**

- \* Adds a library file suffix

- **Parameters**

- \* s - the suffix to add

---

- *buildStatementList*

public List **buildStatementList**( java.io.Reader r, java.lang.String fname )

- **Usage**

- \* Parses a script and creates the associated syntax trees.

- **Parameters**

- \* is - the reader from which the statements are read
  - \* fname - the name of the parsed stream

- **Returns** - list of statements

---

- *defineClass*

public Class **defineClass**( java.lang.String name, byte [] code )

- **Usage**

- \* Converts an array of bytes into an instance of the class Class

- **Exceptions**

- \* java.lang.ClassFormatError - if the class cannot be defined

---

- *defineVariable*

public void **defineVariable**( java.lang.String name, boolean value )

- **Usage**

- \* Defines a boolean variable in the interpreter environment

- **Parameters**

- \* name - the variable's name
  - \* value - the initial value of the variable

- **Exceptions**

- \* java.lang.IllegalStateException - if name is already defined

---

- *defineVariable*

public void **defineVariable**( java.lang.String name, byte value )

- **Usage**
    - \* Defines a byte variable in the interpreter environment
  - **Parameters**
    - \* **name** - the variable's name
    - \* **value** - the initial value of the variable
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if name is already defined
- 

- *defineVariable*

```
public void defineVariable( java.lang.String name, char value )
```

- **Usage**
    - \* Defines a char variable in the interpreter environment
  - **Parameters**
    - \* **name** - the variable's name
    - \* **value** - the initial value of the variable
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if name is already defined
- 

- *defineVariable*

```
public void defineVariable( java.lang.String name, double value )
```

- **Usage**
    - \* Defines an double variable in the interpreter environment
  - **Parameters**
    - \* **name** - the variable's name
    - \* **value** - the initial value of the variable
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if name is already defined
- 

- *defineVariable*

```
public void defineVariable( java.lang.String name, float value )
```

- **Usage**
    - \* Defines an float variable in the interpreter environment
  - **Parameters**
    - \* **name** - the variable's name
    - \* **value** - the initial value of the variable
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if name is already defined
- 

- *defineVariable*

```
public void defineVariable( java.lang.String name, int value )
```

- **Usage**
  - \* Defines an int variable in the interpreter environment
- **Parameters**
  - \* **name** - the variable's name
  - \* **value** - the initial value of the variable
- **Exceptions**

---

\* java.lang.IllegalStateException - if name is already defined

---

- *defineVariable*

public void **defineVariable**( java.lang.String **name**, long **value** )

- **Usage**

- \* Defines an long variable in the interpreter environment

- **Parameters**

- \* **name** - the variable's name
  - \* **value** - the initial value of the variable

- **Exceptions**

- \* java.lang.IllegalStateException - if name is already defined

---

- *defineVariable*

public void **defineVariable**( java.lang.String **name**, java.lang.Object **value** )

- **Usage**

- \* Defines a variable in the interpreter environment

- **Parameters**

- \* **name** - the variable's name
  - \* **value** - the initial value of the variable

- **Exceptions**

- \* java.lang.IllegalStateException - if name is already defined

---

- *defineVariable*

public void **defineVariable**( java.lang.String **name**, java.lang.Object **value**, java.lang.Class **c** )

- **Usage**

- \* Defines a variable in the interpreter environment

- **Parameters**

- \* **name** - the variable's name
  - \* **value** - the initial value of the variable
  - \* **c** - the variable's type.

- **Exceptions**

- \* java.lang.IllegalStateException - if name is already defined

---

- *defineVariable*

public void **defineVariable**( java.lang.String **name**, short **value** )

- **Usage**

- \* Defines a short variable in the interpreter environment

- **Parameters**

- \* **name** - the variable's name
  - \* **value** - the initial value of the variable

- **Exceptions**

- \* java.lang.IllegalStateException - if name is already defined

---

- *finalize*

protected void **finalize**( )

---

– **Usage**

\* Called before the destruction of the interpreter

---

• *getAccessible*

**public boolean getAccessible( )**

---

• *getClassLoader*

**public ClassLoader getClassLoader( )**

---

– **Usage**

\* Gets the class loader

---

• *getClassNames*

**public Set getClassNames( )**

---

– **Usage**

\* Returns the defined class names

– **Returns** - a set of strings

---

• *getExceptionClass*

**public Class getExceptionClass( )**

---

– **Usage**

\* Returns the class of the execution exception

---

• *getLibraryFinder*

**public LibraryFinder getLibraryFinder( )**

---

– **Usage**

\* Gets the library finder

---

• *getParserFactory*

**public ParserFactory getParserFactory( )**

---

– **Usage**

\* Gets the parser factory

---

• *getVariable*

**public Object getVariable( java.lang.String name )**

---

– **Usage**

\* Gets the value of a variable

– **Parameters**

\* **name** - the variable's name

– **Exceptions**

\* **java.lang.IllegalStateException** - if the variable do not exist

---

• *getVariableClass*

**public Class getVariableClass( java.lang.String name )**

---

– **Usage**

\* Gets the class of a variable

- **Parameters**
  - \* **name** - the variable's name
- **Exceptions**
  - \* `java.lang.IllegalStateException` - if the variable do not exist

---

- *getVariableNames*  
`public Set getVariableNames( )`
  - **Usage**
    - \* Returns the defined variable names
  - **Returns** - a set of strings

---

- *interpret*  
`public Object interpret( java.io.InputStream is, java.lang.String fname )`
  - **Usage**
    - \* Runs the interpreter
  - **Parameters**
    - \* **is** - the input stream from which the statements are read
    - \* **fname** - the name of the parsed stream
  - **Returns** - the result of the evaluation of the last statement

---

- *interpret*  
`public Object interpret( java.util.List statements )`
  - **Usage**
    - \* Runs the interpreter on a statement list.
  - **Parameters**
    - \* **statements** - the statement list to evaluate
    - \* **fname** - the name of the parsed stream
  - **Returns** - the result of the evaluation of the last statement

---

- *interpret*  
`public Object interpret( java.io.Reader r, java.lang.String fname )`
  - **Usage**
    - \* Runs the interpreter
  - **Parameters**
    - \* **is** - the reader from which the statements are read
    - \* **fname** - the name of the parsed stream
  - **Returns** - the result of the evaluation of the last statement

---

- *interpret*  
`public Object interpret( java.lang.String fname )`
  - **Usage**
    - \* Runs the interpreter
  - **Parameters**
    - \* **fname** - the name of a file to interpret
  - **Returns** - the result of the evaluation of the last statement

---

- *interpretArguments*

```
protected Object interpretArguments( java.lang.Class c,
koala.dynamicjava.interpreter.TreeInterpreter.ConstructorParametersDescriptor
cpd, java.lang.Object [] args )
```

- **Usage**

- \* This method is used to implement constructor invocation.

- **Parameters**

- \* **c** - the declaring class of the constructor
    - \* **cpd** - the parameter descriptor
    - \* **args** - the arguments passed to this constructor

- **Returns** - the arguments to give to the 'super' or 'this' constructor followed by the new values of the constructor arguments

---

- *interpretArguments*

```
public static Object interpretArguments( java.lang.String key,
java.lang.Object [] args )
```

- **Usage**

- \* This method is used to implement constructor invocation.

- **Parameters**

- \* **key** - the key used to find the informations about the constructor
    - \* **args** - the arguments passed to this constructor

- **Returns** - the arguments to give to the 'super' or 'this' constructor followed by the new values of the constructor arguments

---

- *interpretMethod*

```
protected Object interpretMethod( java.lang.Class c,
koala.dynamicjava.interpreter.TreeInterpreter.MethodDescriptor md,
java.lang.Object obj, java.lang.Object [] params )
```

- **Usage**

- \* Interprets the body of a method

- **Parameters**

- \* **c** - the declaring class of the method
    - \* **md** - the method descriptor
    - \* **obj** - the object (this)
    - \* **params** - the arguments

---

- *invokeMethod*

```
public static Object invokeMethod( java.lang.String key, java.lang.Object
obj, java.lang.Object [] params )
```

- **Usage**

- \* Interprets the body of a method

- **Parameters**

- \* **key** - the key used to find the body of a method
    - \* **obj** - the object (this)
    - \* **params** - the arguments

---

- *loadClass*

```
public Class loadClass( java.lang.String  name )
```

- **Usage**

- \* Loads an interpreted class

- **Parameters**

- \* **s** - the fully qualified name of the class to load

- **Exceptions**

- \* `java.lang.ClassNotFoundException` - if the class cannot be find

---

- *locationToString*

```
private String locationToString( koala.dynamicjava.tree.Node  node )
```

---

- *registerConstructorArguments*

```
public void registerConstructorArguments( java.lang.String  sig,
java.util.List  params, java.util.List  exprs,
koala.dynamicjava.util.ImportationManager  im )
```

- **Usage**

- \* Registers a constructor arguments

---

- *registerMethod*

```
public void registerMethod( java.lang.String  sig,
koala.dynamicjava.tree.MethodDeclaration  md,
koala.dynamicjava.util.ImportationManager  im )
```

- **Usage**

- \* Registers a method.

- **Parameters**

- \* **sig** - the method's signature
  - \* **md** - the method declaration
  - \* **im** - the importation manager

---

- *setAccessible*

```
public void setAccessible( boolean  accessible )
```

---

- *setVariable*

```
public void setVariable( java.lang.String  name, java.lang.Object  value )
```

- **Usage**

- \* Sets the value of a variable

- **Parameters**

- \* **name** - the variable's name
  - \* **value** - the value of the variable

- **Exceptions**

- \* `java.lang.IllegalStateException` - if the assignment is invalid

## 7.2.45 CLASS *TreeInterpreter.ConstructorParametersDescriptor*

---

Used to store the informations about explicit constructors invocation

DECLARATION

---

```
protected class TreeInterpreter.ConstructorParametersDescriptor
extends java.lang.Object
```

CONSTRUCTORS

---

- *TreeInterpreter.ConstructorParametersDescriptor*  
**TreeInterpreter.ConstructorParametersDescriptor**( java.util.List params,  
java.util.List args, koala.dynamicjava.util.ImportationManager im )
  - **Usage**
    - \* Creates a new descriptor

**7.2.46 CLASS TreeInterpreter.MethodDescriptor**

---

Used to store the informations about dynamically created methods

DECLARATION

---

```
protected class TreeInterpreter.MethodDescriptor
extends java.lang.Object
```

CONSTRUCTORS

---

- *TreeInterpreter.MethodDescriptor*  
**TreeInterpreter.MethodDescriptor**( koala.dynamicjava.tree.MethodDeclaration  
md, koala.dynamicjava.util.ImportationManager im )
  - **Usage**
    - \* Creates a new descriptor

**7.2.47 CLASS TypeChecker**

---

This tree visitor checks the typing rules and loads the classes, fields and methods

DECLARATION

---

```
public class TypeChecker
extends koala.dynamicjava.tree.visitor.VisitorObject
```



## FIELDS

---

- private Context context
  - The context

## CONSTRUCTORS

---

- *TypeChecker*  
`public TypeChecker( koala.dynamicjava.interpreter.context.Context ctx )`
  - **Usage**
    - \* Creates a new name visitor
  - **Parameters**
    - \* `ctx` - the context

## METHODS

---

- *checkAssignmentStaticRules*  
`private static void checkAssignmentStaticRules( java.lang.Class lc,  
java.lang.Class rc, koala.dynamicjava.tree.Node node,  
koala.dynamicjava.tree.Node v )`
  - **Usage**
    - \* Checks the typing rules for an assignment
  - **Parameters**
    - \* `lc` - the class of the left part of an assignment
    - \* `rc` - the class of the right part of an assignment
    - \* `node` - the current node
- *checkCastStaticRules*  
`private static void checkCastStaticRules( java.lang.Class tc,  
java.lang.Class ec, koala.dynamicjava.tree.Node n )`
  - **Usage**
    - \* Checks the typing rules in a cast expression
  - **Parameters**
    - \* `tc` - the target class
    - \* `ec` - the expression class
- *checkEqualityStaticRules*  
`private static void checkEqualityStaticRules( java.lang.Class lc,  
java.lang.Class rc, koala.dynamicjava.tree.Node n )`
  - **Usage**
    - \* Checks the typing rules in an equality operation
  - **Parameters**
    - \* `lc` - the class of the left operand

\* **rc** - the class of the right operand  
 \* **s** - the error message  
 \* **n** - the current node

---

- *checkList*

```
private void checkList( java.util.List l )
```

- **Usage**

- \* Check a list of node

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

- **Usage**

- \* Visits an AddAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

- **Usage**

- \* Visits an AddExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

- **Usage**

- \* Visits an AndExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAccess node )
```

- **Usage**

- \* Visits an ArrayAccess

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

- **Usage**

- \* Visits an ArrayAllocation

- **Parameters**

- \* **node** - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayInitializer node )  
 – **Usage**  
   \* Visits a ArrayInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayType node )  
 – **Usage**  
   \* Visits a ArrayType  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )  
 – **Usage**  
   \* Visits a BitAndAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )  
 – **Usage**  
   \* Visits a BitOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )  
 – **Usage**  
   \* Visits a BlockStatement  
 – **Parameters**  
   \* node - the node to visit  


---

- **Usage**
    - \* Visits a BlockStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.CastExpression **node** )
  - **Usage**
    - \* Visits a CastExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.CatchStatement **node** )
  - **Usage**
    - \* Visits a CatchStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ClassAllocation **node** )
  - **Usage**
    - \* Visits a ClassAllocation
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ComplementExpression **node** )
  - **Usage**
    - \* Visits a ComplementExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ConditionalExpression **node** )
  - **Usage**
    - \* Visits a ConditionalExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.DivideAssignExpression **node** )
  - **Usage**
    - \* Visits an DivideAssignExpression

- **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideExpression node )
```

    - **Usage**
      - \* Visits a DivideExpression
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.DoStatement node )
```

      - **Usage**
        - \* Visits a DoStatement
      - **Parameters**
        - \* `node` - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.EqualExpression node )
```

        - **Usage**
          - \* Visits an EqualExpression
        - **Parameters**
          - \* `node` - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
```

          - **Usage**
            - \* Visits a ExclusiveOrAssignExpression
          - **Parameters**
            - \* `node` - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
```

            - **Usage**
              - \* Visits a ExclusiveOrExpression
            - **Parameters**
              - \* `node` - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.FormalParameter node )
```

              - **Usage**
                - \* Visits a FormalParameter
              - **Parameters**

- \* **node** - the node to visit
  - **Returns** - the class of the parameter

---
- *visit*
  - public Object **visit**( koala.dynamicjava.tree.ForStatement **node** )
  - **Usage**
    - \* Visits a ForStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*
  - public Object **visit**( koala.dynamicjava.tree.FunctionCall **node** )
  - **Usage**
    - \* Visits a FunctionCall
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*
  - public Object **visit**( koala.dynamicjava.tree.GreaterExpression **node** )
  - **Usage**
    - \* Visits a GreaterExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*
  - public Object **visit**( koala.dynamicjava.tree.GreaterOrEqualExpression **node** )
  - **Usage**
    - \* Visits a GreaterOrEqualExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*
  - public Object **visit**( koala.dynamicjava.tree.IfThenElseStatement **node** )
  - **Usage**
    - \* Visits an IfThenElseStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*
  - public Object **visit**( koala.dynamicjava.tree.IfThenStatement **node** )
  - **Usage**
    - \* Visits an IfThenStatement
  - **Parameters**
    - \* **node** - the node to visit

---

- *visit*  
public Object visit( koala.dynamicjava.tree.ImportDeclaration node )
  - Usage
    - \* Visits an ImportDeclaration
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.InnerAllocation node )
  - Usage
    - \* Visits a InnerAllocation
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
  - Usage
    - \* Visits a InstanceOfExpression
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.LabeledStatement node )
  - Usage
    - \* Visits a LabeledStatement
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.LessExpression node )
  - Usage
    - \* Visits a LessExpression
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
  - Usage
    - \* Visits a LessOrEqualExpression
  - Parameters
    - \* node - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.Literal node )

- **Usage**
    - \* Visits a Literal
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
```

  - **Usage**
    - \* Visits a MethodDeclaration
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MinusExpression node )
```

  - **Usage**
    - \* Visits a MinusExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

  - **Usage**
    - \* Visits an MultiplyAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

  - **Usage**
    - \* Visits a MultiplyExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

  - **Usage**
    - \* Visits an NotEqualExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotExpression node )
```

  - **Usage**
    - \* Visits a NotExpression



- **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
```

    - **Usage**
      - \* Visits an ObjectFieldAccess
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
```

      - **Usage**
        - \* Visits an ObjectMethodCall
      - **Parameters**
        - \* `node` - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.OrExpression node )
```

        - **Usage**
          - \* Visits an OrExpression
        - **Parameters**
          - \* `node` - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
```

          - **Usage**
            - \* Visits a PackageDeclaration
          - **Parameters**
            - \* `node` - the node to visit
          - **Returns** - null

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.PlusExpression node )
```

            - **Usage**
              - \* Visits a PlusExpression
            - **Parameters**
              - \* `node` - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement node )
```

              - **Usage**
                - \* Visits a PostDecrement
              - **Parameters**

\* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.PostIncrement node )

- Usage

- \* Visits a PostIncrement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.PreDecrement node )

- Usage

- \* Visits a PreDecrement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.PreIncrement node )

- Usage

- \* Visits a PreIncrement

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.PrimitiveType node )

- Usage

- \* Visits a PrimitiveType

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.QualifiedName node )

- Usage

- \* Visits a QualifiedName

- Parameters

- \* node - the node to visit

---

- *visit*

public Object visit( koala.dynamicjava.tree.ReferenceType node )

- Usage

- \* Visits a ReferenceType

- Parameters

- \* node - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
```

- **Usage**

- \* Visits an RemainderAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderExpression node )
```

- **Usage**

- \* Visits a RemainderExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ReturnStatement node )
```

- **Usage**

- \* Visits a ReturnStatement

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
```

- **Usage**

- \* Visits a ShiftLeftAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
```

- **Usage**

- \* Visits a ShiftLeftExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
```

- **Usage**

- \* Visits a ShiftRightAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightExpression  node )
```

– **Usage**

\* Visits a ShiftRightExpression

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAllocation  node )
```

– **Usage**

\* Visits a SimpleAllocation

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAssignExpression  node )
```

– **Usage**

\* Visits a SimpleAssignExpression

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.StaticFieldAccess  node )
```

– **Usage**

\* Visits a StaticFieldAccess

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.StaticMethodCall  node )
```

– **Usage**

\* Visits a StaticMethodCall

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractAssignExpression  node )
```

– **Usage**

\* Visits an SubtractAssignExpression

– **Parameters**

\* **node** - the node to visit

---

• *visit*

```
public Object visit( koala.dynamicjava.tree.SubtractExpression  node )
```

- **Usage**
    - \* Visits a SubtractExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperFieldAccess  node )
```

  - **Usage**
    - \* Visits a SuperFieldAccess
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperMethodCall  node )
```

  - **Usage**
    - \* Visits a SuperMethodCall
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchBlock  node )
```

  - **Usage**
    - \* Visits a SwitchBlock
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement  node )
```

  - **Usage**
    - \* Visits a SwitchStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement  node )
```

  - **Usage**
    - \* Visits a SynchronizedStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement  node )
```

  - **Usage**
    - \* Visits a ThrowStatement

- **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement  node )
```

    - **Usage**
      - \* Visits a TryStatement
    - **Parameters**
      - \* **node** - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression  node )
```

      - **Usage**
        - \* Visits a TypeExpression
      - **Parameters**
        - \* **node** - the node to visit

---
    - *visit*

```
public Object visit(
koala.dynamicjava.tree.UnsignedShiftRightAssignExpression  node )
```

        - **Usage**
          - \* Visits a UnsignedShiftRightAssignExpression
        - **Parameters**
          - \* **node** - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression
node )
```

          - **Usage**
            - \* Visits a UnsignedShiftRightExpression
          - **Parameters**
            - \* **node** - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration  node )
```

            - **Usage**
              - \* Visits a VariableDeclaration
            - **Parameters**
              - \* **node** - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.WhileStatement  node )
```

              - **Usage**
                - \* Visits a WhileStatement
              - **Parameters**

\* node - the node to visit

---

- *visitBitwiseAssign*

```
private Class visitBitwiseAssign( koala.dynamicjava.tree.BinaryExpression
node )
```

– Usage

\* Checks a bitwise expression

---

- *visitBitwiseExpression*

```
private Class visitBitwiseExpression( koala.dynamicjava.tree.BinaryExpression
node )
```

– Usage

\* Visits a bitwise expression

---

- *visitNumericExpression*

```
private static Class visitNumericExpression(
koala.dynamicjava.tree.BinaryExpression node, java.lang.String s )
```

– Usage

\* Visits a numeric expression

---

- *visitRelationalExpression*

```
private Class visitRelationalExpression(
koala.dynamicjava.tree.BinaryExpression node )
```

– Usage

\* Visits a relational expression

---

- *visitShiftExpression*

```
private Class visitShiftExpression( koala.dynamicjava.tree.BinaryExpression
node )
```

– Usage

\* Visits a shift expression

---

- *visitUnaryOperation*

```
private Class visitUnaryOperation( koala.dynamicjava.tree.UnaryExpression
node, java.lang.String s )
```

– Usage

\* Visits an unary operation

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.visitor.VisitorObject

---

( in 5.2.1, page 160)

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

– Usage

\* Visits an AddAssignExpression

- **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

  - **Usage**
  - \* Visits a `AddExpression`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

  - **Usage**
  - \* Visits a `AndExpression`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAccess node )
```

  - **Usage**
  - \* Visits a `ArrayAccess`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

  - **Usage**
  - \* Visits an `ArrayAllocation`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayInitializer node )
```

  - **Usage**
  - \* Visits an `ArrayInitializer`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayType node )
```

  - **Usage**
  - \* Visits a `ArrayType`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )
```

  - **Usage**
  - \* Visits a `BitAndAssignExpression`
  - **Parameters**
  - \* `node` - the node to visit

---



- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )  
 – **Usage**  
   \* Visits a BitOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )  
 – **Usage**  
   \* Visits a BlockStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BreakStatement node )  
 – **Usage**  
   \* Visits a BreakStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.CastExpression node )  
 – **Usage**  
   \* Visits a CastExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.CatchStatement node )  
 – **Usage**  
   \* Visits a CatchStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassAllocation node )

- **Usage**
    - \* Visits an ClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration node )
```

  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer node )
```

  - **Usage**
    - \* Visits a ClassInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression node )
```

  - **Usage**
    - \* Visits a ComplementExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
```

  - **Usage**
    - \* Visits a ConditionalExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )
```

  - **Usage**
    - \* Visits a ConstructorDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )
```

  - **Usage**
    - \* Visits a ConstructorInvocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ContinueStatement node )
```

  - **Usage**
    - \* Visits a ContinueStatement
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
      - Usage
        - \* Visits an DivideAssignExpression
      - Parameters
        - \* node - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.DivideExpression node )
      - Usage
        - \* Visits a DivideExpression
      - Parameters
        - \* node - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.DoStatement node )
      - Usage
        - \* Visits a DoStatement
      - Parameters
        - \* node - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.EmptyStatement node )
      - Usage
        - \* Visits an EmptyStatement
      - Parameters
        - \* node - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.EqualExpression node )
      - Usage
        - \* Visits a EqualExpression
      - Parameters
        - \* node - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
      - Usage
        - \* Visits a ExclusiveOrAssignExpression
      - Parameters
        - \* node - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
      - Usage
        - \* Visits a ExclusiveOrExpression
      - Parameters
        - \* node - the node to visit
-

- *visit*  
 public Object visit( koala.dynamicjava.tree.FieldDeclaration node )  
 – **Usage**  
   \* Visits a FieldDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FormalParameter node )  
 – **Usage**  
   \* Visits a FormalParameter  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ForStatement node )  
 – **Usage**  
   \* Visits a ForStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FunctionCall node )  
 – **Usage**  
   \* Visits a FunctionCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterExpression node )  
 – **Usage**  
   \* Visits a GreaterExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )  
 – **Usage**  
   \* Visits a GreaterOrEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )  
 – **Usage**  
   \* Visits a IfThenElseStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenStatement node )

- **Usage**
    - \* Visits a IfThenStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration node )
```

  - **Usage**
    - \* Visits an ImportDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerAllocation node )
```

  - **Usage**
    - \* Visits an InnerAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )
```

  - **Usage**
    - \* Visits an InnerClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceInitializer node )
```

  - **Usage**
    - \* Visits a InstanceInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
```

  - **Usage**
    - \* Visits an InstanceOfExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )
```

  - **Usage**
    - \* Visits a InterfaceDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.LabeledStatement node )
```

  - **Usage**
    - \* Visits a LabeledStatement
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )  
 – Usage  
   \* Visits a LessExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )  
 – Usage  
   \* Visits a LessOrEqualExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.Literal node )  
 – Usage  
   \* Visits a Literal  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.MethodDeclaration node )  
 – Usage  
   \* Visits a MethodDeclaration  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.MinusExpression node )  
 – Usage  
   \* Visits a MinusExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )  
 – Usage  
   \* Visits an MultiplyAssignExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.MultiplyExpression node )  
 – Usage  
   \* Visits a MultiplyExpression  
 – Parameters  
   \* node - the node to visit
-

- *visit*  
 public Object visit( koala.dynamicjava.tree.NotEqualExpression node )  
 – **Usage**  
   \* Visits a NotEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotExpression node )  
 – **Usage**  
   \* Visits a NotExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )  
 – **Usage**  
   \* Visits a ObjectFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )  
 – **Usage**  
   \* Visits a ObjectMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.OrExpression node )  
 – **Usage**  
   \* Visits a OrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PackageDeclaration node )  
 – **Usage**  
   \* Visits an PackageDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PlusExpression node )  
 – **Usage**  
   \* Visits a PlusExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PostDecrement node )

- **Usage**
    - \* Visits a PostDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement node )
```

  - **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.QualifiedName node )
```

  - **Usage**
    - \* Visits a QualifiedName
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReferenceType node )
```

  - **Usage**
    - \* Visits a ReferenceType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
```

  - **Usage**
    - \* Visits an RemainderAssignExpression
  - **Parameters**



- 
- \* node - the node to visit
  - - *visit*
    - public Object visit( koala.dynamicjava.tree.RemainderExpression node )
    - Usage
      - \* Visits a RemainderExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ReturnStatement node )
    - Usage
      - \* Visits a ReturnStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
    - Usage
      - \* Visits an ShiftLeftAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
    - Usage
      - \* Visits a ShiftLeftExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
    - Usage
      - \* Visits an ShiftRightAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
    - Usage
      - \* Visits a ShiftRightExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
    - Usage
      - \* Visits an SimpleAllocation
      - Parameters
      - \* node - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )
```

  - **Usage**  
 \* Visits an SimpleAssignExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )
```

  - **Usage**  
 \* Visits a StaticFieldAccess
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.StaticMethodCall node )
```

  - **Usage**  
 \* Visits a StaticMethodCall
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
```

  - **Usage**  
 \* Visits an SubtractAssignExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.SubtractExpression node )
```

  - **Usage**  
 \* Visits a SubtractExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )
```

  - **Usage**  
 \* Visits a SuperFieldAccess
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.SuperMethodCall node )
```

  - **Usage**  
 \* Visits a SuperMethodCall
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.SwitchBlock node )
```

- **Usage**
    - \* Visits a SwitchBlock
  - **Parameters**
    - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement node )
```

    - **Usage**
      - \* Visits a SwitchStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )
```

    - **Usage**
      - \* Visits a SynchronizedStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression node )
```

    - **Usage**
      - \* Visits a ThisExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement node )
```

    - **Usage**
      - \* Visits a ThrowStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement node )
```

    - **Usage**
      - \* Visits a TryStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression node )
```

    - **Usage**
      - \* Visits a TypeExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )
```

    - **Usage**
      - \* Visits an UnsignedShiftRightAssignExpression

- **Parameters**
    - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression  node )
```

    - **Usage**
      - \* Visits a UnsignedShiftRightExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration  node )
```

    - **Usage**
      - \* Visits a VariableDeclaration
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.WhileStatement  node )
```

    - **Usage**
      - \* Visits a WhileStatement
    - **Parameters**
      - \* **node** - the node to visit

## 7.2.48 CLASS UninitializedObject

---

The INSTANCE object of this class represents an uninitialized object

### DECLARATION

---

```
public class UninitializedObject
extends java.lang.Object
```

### FIELDS

---

- public static UninitializedObject INSTANCE
  - The only instance of this class

### CONSTRUCTORS

---

- *UninitializedObject*

```
private UninitializedObject( )
```

  - **Usage**
    - \* Private creation

## Chapter 8

# Package koala.dynamicjava.util

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Classes</b>	
<b>AmbiguousFieldException</b> .....	573
<i>This exception is thrown when a lookup for a field is ambiguous.</i>	
<b>BufferedImportationManager</b> .....	574
<i>A buffered version of the importation manager</i>	
<b>Constants</b> .....	577
<i>This class contains constants</i>	
<b>DisplayVisitor</b> .....	578
<i>This tree visitor displays the nodes of the tree on a given stream</i>	
<b>FileFinder</b> .....	606
<i>This class represents an object that manages a set of path where to find files.</i>	
<b>ImportationManager</b> .....	607
<i>The instances of this class manages the importation clauses.</i>	
<b>LibraryFinder</b> .....	610
<i>The instances of LibraryFinder are used to locate files with given suffixes.</i>	
<b>LocalizedMessageReader</b> .....	611
<i>The instances of this class read localized messages in resource files.</i>	
<b>ReflectionUtilities</b> .....	612
<i>This class contains a collection of utility methods for reflection.</i>	
<hr/>	

## 8.1 Classes

### 8.1.1 CLASS AmbiguousFieldException

This exception is thrown when a lookup for a field is ambiguous.

#### DECLARATION

```
public class AmbiguousFieldException
extends java.lang.Exception
```

#### CONSTRUCTORS

- *AmbiguousFieldException*  
 public **AmbiguousFieldException**( )  
 – **Usage**  
 \* Constructs an **AmbiguousFieldException** with no detail message.

---

- *AmbiguousFieldException*  
 public **AmbiguousFieldException**( java.lang.String s )  
 – **Usage**  
 \* Constructs an **AmbiguousFieldException** with the specified detail message.  
 – **Parameters**  
 \* **s** - the detail message (a key in a resource file).

#### METHODS INHERITED FROM CLASS java.lang.Exception

#### METHODS INHERITED FROM CLASS java.lang.Throwable

- *fillInStackTrace*  
public synchronized native Throwable fillInStackTrace( )
- *getCause*  
public Throwable getCause( )
- *getLocalizedMessage*  
public String getLocalizedMessage( )
- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )

- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int    )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable    )
- *printStackTrace*  
public void printStackTrace(    )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream    )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter    )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream    ,  
java.lang.StackTraceElement []    )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter    ,  
java.lang.StackTraceElement []    )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement []    )
- *toString*  
public String toString(    )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream    )

### 8.1.2 CLASS *BufferedImportationManager*

---

A buffered version of the importation manager

#### DECLARATION

---

```
public class BufferedImportationManager
extends koala.dynamicjava.util.ImportationManager
```

#### CONSTRUCTORS

---

- *BufferedImportationManager*  
public **BufferedImportationManager**( java.lang.ClassLoader    cl )
  - **Usage**
    - \* Creates a new importation manager.
  - **Parameters**
    - \* cl - the class loader to use
- *BufferedImportationManager*  
protected **BufferedImportationManager**(  
koala.dynamicjava.util.ImportationManager    im )
  - **Usage**
    - \* Copy constructor

METHODS

---

- *clone*  
`public Object clone( )`
  - **Usage**
    - \* Returns a copy of this object

---
- *declareClassImport*  
`public void declareClassImport( java.lang.String cname )`
  - **Usage**
    - \* Declares a new single-type-import clause
  - **Parameters**
    - \* `cname` - the fully qualified class name
  - **Exceptions**
    - \* `java.lang.ClassNotFoundException` - if the class cannot be found

---
- *declarePackageImport*  
`public void declarePackageImport( java.lang.String pkg )`
  - **Usage**
    - \* Declares a new import-on-demand clause
  - **Parameters**
    - \* `pkg` - the package name

---
- *lookupClass*  
`public Class lookupClass( java.lang.String cname, java.lang.String ccname )`
  - **Usage**
    - \* Loads the class that match to the given name in the source file
  - **Parameters**
    - \* `cname` - the name of the class to find
    - \* `ccname` - the name of the current class or null
  - **Returns** - the class found
  - **Exceptions**
    - \* `java.lang.ClassNotFoundException` - if the class cannot be loaded

---
- *setCurrentPackage*  
`public void setCurrentPackage( java.lang.String pkg )`
  - **Usage**
    - \* Sets the current package. This has no influence on the behaviour of the `lookupClass` method.
  - **Parameters**
    - \* `pkg` - the package name



METHODS INHERITED FROM CLASS `koala.dynamicjava.util.ImportationManager`

---

( in 8.1.6, page 607)

- *clone*  
`public Object clone( )`  
  - **Usage**  
 \* Returns a copy of this object
- *declareClassImport*  
`public void declareClassImport( java.lang.String cname )`  
  - **Usage**  
 \* Declares a new single-type-import clause
  - **Parameters**  
 \* `cname` - the fully qualified class name
  - **Exceptions**  
 \* `java.lang.ClassNotFoundException` - if the class cannot be found
- *declarePackageImport*  
`public void declarePackageImport( java.lang.String pkg )`  
  - **Usage**  
 \* Declares a new import-on-demand clause
  - **Parameters**  
 \* `pkg` - the package name
- *findInnerClass*  
`protected Class findInnerClass( java.lang.String s )`  
  - **Usage**  
 \* Searches for an inner class from its name in the dotted notation
- *getCurrentPackage*  
`public String getCurrentPackage( )`  
  - **Usage**  
 \* Returns the current package
- *getImportOnDemandClauses*  
`public List getImportOnDemandClauses( )`  
  - **Usage**  
 \* Returns the import-on-demand clauses
- *getOuterNames*  
`protected List getOuterNames( java.lang.String cname )`  
  - **Usage**  
 \* Returns a list of the outer classes names
- *getSingleTypeImportClauses*  
`public List getSingleTypeImportClauses( )`  
  - **Usage**  
 \* Returns the single-type-import clauses
- *hasSuffix*  
`protected boolean hasSuffix( java.lang.String c1, java.lang.String c2 )`  
  - **Usage**

- \* Tests whether the fully qualified class name c1 ends with c2
- 
- *lookupClass*  
 public Class lookupClass( java.lang.String cname, java.lang.String ccname )
    - **Usage**
      - \* Loads the class that match to the given name in the source file
    - **Parameters**
      - \* **cname** - the name of the class to find
      - \* **ccname** - the name of the current class or null
    - **Returns** - the class found
    - **Exceptions**
      - \* java.lang.ClassNotFoundException - if the class cannot be loaded
- 
- *setClassLoader*  
 public void setClassLoader( java.lang.ClassLoader cl )
    - **Usage**
      - \* Sets the class loader
- 
- *setCurrentPackage*  
 public void setCurrentPackage( java.lang.String pkg )
    - **Usage**
      - \* Sets the current package. This has no influence on the behaviour of the lookupClass method.
    - **Parameters**
      - \* **pkg** - the package name

### 8.1.3 CLASS Constants

---

This class contains constants

#### DECLARATION

---

```
public class Constants
extends java.lang.Object
```

#### FIELDS

---

- public static final Object EMPTY\_OBJECT\_ARRAY
  - An empty object array
- public static final Class EMPTY\_CLASS\_ARRAY
  - An empty class array

---

## CONSTRUCTORS

- *Constants*

**protected Constants( )**

- **Usage**

\* This class has not to be instantiated

### 8.1.4 CLASS DisplayVisitor

---

This tree visitor displays the nodes of the tree on a given stream

## DECLARATION

---

```
public class DisplayVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

## FIELDS

---

- private PrintStream out
  - The output stream
- private String indentation
  - The current indentation

## CONSTRUCTORS

---

- *DisplayVisitor*

**public DisplayVisitor( java.io.OutputStream os )**

- **Usage**

\* Creates a new display visitor

- **Parameters**

\* **os** - the output tree

## METHODS

---

- *displayBinary*

**private void displayBinary( koala.dynamicjava.tree.BinaryExpression be )**

- **Usage**

\* Displays a binary expression

- 
- *displayProperties*

**private void displayProperties( koala.dynamicjava.tree.Node node )**

- **Usage**
    - \* Displays the properties of a node

---
- *displayUnary*  
`private void displayUnary( koala.dynamicjava.tree.UnaryExpression ue )`
    - **Usage**
      - \* Displays an unary expression

---
  - *indent*  
`private void indent( )`
    - **Usage**
      - \* Adds a level of indentation

---
  - *print*  
`private void print( java.lang.String s )`
    - **Usage**
      - \* Prints an indented line

---
  - *unindent*  
`private void unindent( )`
    - **Usage**
      - \* Removes a level of indentation

---
  - *visit*  
`public Object visit( koala.dynamicjava.tree.AddAssignExpression node )`
    - **Usage**
      - \* Visits an AddAssignExpression
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*  
`public Object visit( koala.dynamicjava.tree.AddExpression node )`
    - **Usage**
      - \* Visits a AddExpression
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*  
`public Object visit( koala.dynamicjava.tree.AndExpression node )`
    - **Usage**
      - \* Visits an AndExpression
    - **Parameters**
      - \* `node` - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAccess node )  
 – **Usage**  
   \* Visits a ArrayAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayAllocation node )  
 – **Usage**  
   \* Visits an ArrayAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayInitializer node )  
 – **Usage**  
   \* Visits a ArrayInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayType node )  
 – **Usage**  
   \* Visits a ArrayType  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )  
 – **Usage**  
   \* Visits a BitAndAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )

- **Usage**
    - \* Visits a BitOrAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrExpression node )
```

    - **Usage**
      - \* Visits a BitOrExpression
    - **Parameters**
      - \* **node** - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.BlockStatement node )
```

      - **Usage**
        - \* Visits a BlockStatement
      - **Parameters**
        - \* **node** - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.BreakStatement node )
```

        - **Usage**
          - \* Visits a BreakStatement
        - **Parameters**
          - \* **node** - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.CastExpression node )
```

          - **Usage**
            - \* Visits a CastExpression
          - **Parameters**
            - \* **node** - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.CatchStatement node )
```

            - **Usage**
              - \* Visits a CatchStatement
            - **Parameters**
              - \* **node** - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.ClassAllocation node )
```

              - **Usage**
                - \* Visits an ClassAllocation

- **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration node )
```

    - **Usage**
      - \* Visits a ClassDeclaration
    - **Parameters**
      - \* `node` - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer node )
```

      - **Usage**
        - \* Visits a ClassInitializer
      - **Parameters**
        - \* `node` - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression node )
```

        - **Usage**
          - \* Visits a ComplementExpression
        - **Parameters**
          - \* `node` - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
```

          - **Usage**
            - \* Visits a ConditionalExpression
          - **Parameters**
            - \* `node` - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )
```

            - **Usage**
              - \* Visits a ConstructorDeclaration
            - **Parameters**
              - \* `node` - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )
```

              - **Usage**
                - \* Visits a ConstructorInvocation
              - **Parameters**
                - \* `node` - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ContinueStatement node )
```

- **Usage**

- \* Visits a ContinueStatement

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
```

- **Usage**

- \* Visits an DivideAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideExpression node )
```

- **Usage**

- \* Visits a DivideExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.DoStatement node )
```

- **Usage**

- \* Visits a DoStatement

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.EmptyStatement node )
```

- **Usage**

- \* Visits an EmptyStatement

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.EqualExpression node )
```

- **Usage**

- \* Visits a EqualExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
```



- **Usage**
    - \* Visits a ExclusiveOrAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
```

    - **Usage**
      - \* Visits a ExclusiveOrExpression
    - **Parameters**
      - \* **node** - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.FieldDeclaration node )
```

      - **Usage**
        - \* Visits a FieldDeclaration
      - **Parameters**
        - \* **node** - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.FormalParameter node )
```

        - **Usage**
          - \* Visits a FormalParameter
        - **Parameters**
          - \* **node** - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.ForStatement node )
```

          - **Usage**
            - \* Visits a ForStatement
          - **Parameters**
            - \* **node** - the node to visit

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.FunctionCall node )
```

            - **Usage**
              - \* Visits a FunctionCall
            - **Parameters**
              - \* **node** - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterExpression node )
```

              - **Usage**
                - \* Visits a GreaterExpression

---

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )`

– **Usage**

\* Visits a GreaterOrEqualExpression

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )`

– **Usage**

\* Visits an IfThenElseStatement

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.IfThenStatement node )`

– **Usage**

\* Visits an IfThenStatement

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.ImportDeclaration node )`

– **Usage**

\* Visits an ImportDeclaration

– **Parameters**

\* **node** - the node to visit

– **Returns** - null

---

• *visit*

`public Object visit( koala.dynamicjava.tree.InnerAllocation node )`

– **Usage**

\* Visits an InnerAllocation

– **Parameters**

\* **node** - the node to visit

---

• *visit*

`public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )`

– **Usage**

\* Visits an InnerClassAllocation

– **Parameters**

- 
- \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceInitializer node )  
 – Usage  
   \* Visits a InstanceInitializer  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )  
 – Usage  
   \* Visits a InstanceOfExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )  
 – Usage  
   \* Visits an InterfaceDeclaration  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LabeledStatement node )  
 – Usage  
   \* Visits a LabeledStatement  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )  
 – Usage  
   \* Visits a LessExpression  
 – Parameters  
   \* node - the node to visit
- 
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )  
 – Usage  
   \* Visits a LessOrEqualExpression  
 – Parameters  
   \* node - the node to visit
-

- *visit*

```
public Object visit( koala.dynamicjava.tree.Literal  node )
```

- **Usage**

- \* Visits a Literal

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.MethodDeclaration  node )
```

- **Usage**

- \* Visits a MethodDeclaration

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.MinusExpression  node )
```

- **Usage**

- \* Visits a MinusExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression  node )
```

- **Usage**

- \* Visits an MultiplyAssignExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression  node )
```

- **Usage**

- \* Visits a MultiplyExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression  node )
```

- **Usage**

- \* Visits a NotEqualExpression

- **Parameters**

- \* **node** - the node to visit

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.NotExpression  node )
```

- **Usage**
    - \* Visits a NotExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess  node )
```

    - **Usage**
      - \* Visits an ObjectFieldAccess
    - **Parameters**
      - \* **node** - the node to visit

---
  - *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall  node )
```

      - **Usage**
        - \* Visits a ObjectMethodCall
      - **Parameters**
        - \* **node** - the node to visit

---
    - *visit*

```
public Object visit( koala.dynamicjava.tree.OrExpression  node )
```

        - **Usage**
          - \* Visits an OrExpression
        - **Parameters**
          - \* **node** - the node to visit

---
      - *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration  node )
```

          - **Usage**
            - \* Visits a PackageDeclaration
          - **Parameters**
            - \* **node** - the node to visit
          - **Returns** - null

---
        - *visit*

```
public Object visit( koala.dynamicjava.tree.PlusExpression  node )
```

            - **Usage**
              - \* Visits a PlusExpression
            - **Parameters**
              - \* **node** - the node to visit

---
          - *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement  node )
```

              - **Usage**

- \* Visits a PostDecrement
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.PostIncrement node )`
  - **Usage**
  - \* Visits a PostIncrement
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.PreDecrement node )`
  - **Usage**
  - \* Visits a PreDecrement
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.PreIncrement node )`
  - **Usage**
  - \* Visits a PreIncrement
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.PrimitiveType node )`
  - **Usage**
  - \* Visits a PrimitiveType
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.QualifiedName node )`
  - **Usage**
  - \* Visits a QualifiedName
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.ReferenceType node )`
  - **Usage**
  - \* Visits a ReferenceType
  - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
```

    - **Usage**
      - \* Visits an RemainderAssignExpression
    - **Parameters**
      - \* node - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderExpression node )
```

    - **Usage**
      - \* Visits a RemainderExpression
    - **Parameters**
      - \* node - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReturnStatement node )
```

    - **Usage**
      - \* Visits a ReturnStatement
    - **Parameters**
      - \* node - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
```

    - **Usage**
      - \* Visits an ShiftLeftAssignExpression
    - **Parameters**
      - \* node - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
```

    - **Usage**
      - \* Visits a ShiftLeftExpression
    - **Parameters**
      - \* node - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
```

    - **Usage**
      - \* Visits an ShiftRightAssignExpression
    - **Parameters**

- 
- \* node - the node to visit
- 
- *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
    - Usage
      - \* Visits a ShiftRightExpression
    - Parameters
      - \* node - the node to visit
- 
- *visit*
    - public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
    - Usage
      - \* Visits an SimpleAllocation
    - Parameters
      - \* node - the node to visit
- 
- *visit*
    - public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )
    - Usage
      - \* Visits an SimpleAssignExpression
    - Parameters
      - \* node - the node to visit
- 
- *visit*
    - public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )
    - Usage
      - \* Visits a StaticFieldAccess
    - Parameters
      - \* node - the node to visit
- 
- *visit*
    - public Object visit( koala.dynamicjava.tree.StaticMethodCall node )
    - Usage
      - \* Visits a StaticMethodCall
    - Parameters
      - \* node - the node to visit
- 
- *visit*
    - public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
    - Usage
      - \* Visits an SubtractAssignExpression
    - Parameters
      - \* node - the node to visit
-



- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractExpression node )  
 – **Usage**  
   \* Visits a SubtractExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )  
 – **Usage**  
   \* Visits a SuperFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperMethodCall node )  
 – **Usage**  
   \* Visits a SuperMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SwitchBlock node )  
 – **Usage**  
   \* Visits a SwitchBlock  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SwitchStatement node )  
 – **Usage**  
   \* Visits a SwitchStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )  
 – **Usage**  
   \* Visits a SynchronizedStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ThisExpression node )  


---

- **Usage**
    - \* Visits a ThisExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement node )
```

  - **Usage**
    - \* Visits a ThrowStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement node )
```

  - **Usage**
    - \* Visits a TryStatement
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression node )
```

  - **Usage**
    - \* Visits a TypeExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit(
koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )
```

  - **Usage**
    - \* Visits an UnsignedShiftRightAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression
node )
```

  - **Usage**
    - \* Visits a UnsignedShiftRightExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration node )
```

  - **Usage**

- \* Visits a VariableDeclaration
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.WhileStatement node )
```

  - **Usage**
    - \* Visits a WhileStatement
  - **Parameters**
    - \* **node** - the node to visit

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.visitor.VisitorObject

---

( in 5.2.1, page 160)

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

  - **Usage**
    - \* Visits an AddAssignExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

  - **Usage**
    - \* Visits a AddExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

  - **Usage**
    - \* Visits a AndExpression
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAccess node )
```

  - **Usage**
    - \* Visits a ArrayAccess
  - **Parameters**
    - \* **node** - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

  - **Usage**
    - \* Visits an ArrayAllocation
  - **Parameters**
    - \* **node** - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayInitializer node )  
 – **Usage**  
   \* Visits an ArrayInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ArrayType node )  
 – **Usage**  
   \* Visits a ArrayType  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )  
 – **Usage**  
   \* Visits a BitAndAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )  
 – **Usage**  
   \* Visits a BitOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )  
 – **Usage**  
   \* Visits a BlockStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BreakStatement node )

- **Usage**
    - \* Visits a BreakStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.CastExpression node )
```

  - **Usage**
    - \* Visits a CastExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.CatchStatement node )
```

  - **Usage**
    - \* Visits a CatchStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassAllocation node )
```

  - **Usage**
    - \* Visits an ClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration node )
```

  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer node )
```

  - **Usage**
    - \* Visits a ClassInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression node )
```

  - **Usage**
    - \* Visits a ComplementExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
```

  - **Usage**
    - \* Visits a ConditionalExpression
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )
      - Usage
        - \* Visits a ConstructorDeclaration
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )
      - Usage
        - \* Visits a ConstructorInvocation
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ContinueStatement node )
      - Usage
        - \* Visits a ContinueStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
      - Usage
        - \* Visits an DivideAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.DivideExpression node )
      - Usage
        - \* Visits a DivideExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.DoStatement node )
      - Usage
        - \* Visits a DoStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.EmptyStatement node )
      - Usage
        - \* Visits an EmptyStatement
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.EqualExpression node )  
 – **Usage**  
   \* Visits a EqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )  
 – **Usage**  
   \* Visits a ExclusiveOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )  
 – **Usage**  
   \* Visits a ExclusiveOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FieldDeclaration node )  
 – **Usage**  
   \* Visits a FieldDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FormalParameter node )  
 – **Usage**  
   \* Visits a FormalParameter  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ForStatement node )  
 – **Usage**  
   \* Visits a ForStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FunctionCall node )  
 – **Usage**  
   \* Visits a FunctionCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterExpression node )

- **Usage**
    - \* Visits a GreaterExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression  node )
```

  - **Usage**
    - \* Visits a GreaterOrEqualExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenElseStatement  node )
```

  - **Usage**
    - \* Visits a IfThenElseStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.IfThenStatement  node )
```

  - **Usage**
    - \* Visits a IfThenStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration  node )
```

  - **Usage**
    - \* Visits an ImportDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerAllocation  node )
```

  - **Usage**
    - \* Visits an InnerAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerClassAllocation  node )
```

  - **Usage**
    - \* Visits an InnerClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceInitializer  node )
```

  - **Usage**
    - \* Visits a InstanceInitializer
  - **Parameters**



- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
      - Usage
        - \* Visits an InstanceOfExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )
      - Usage
        - \* Visits a InterfaceDeclaration
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.LabeledStatement node )
      - Usage
        - \* Visits a LabeledStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )
      - Usage
        - \* Visits a LessExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
      - Usage
        - \* Visits a LessOrEqualExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.Literal node )
      - Usage
        - \* Visits a Literal
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
      - Usage
        - \* Visits a MethodDeclaration
      - Parameters
        - \* node - the node to visit

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.MinusExpression node )
```

  - **Usage**  
 \* Visits a MinusExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

  - **Usage**  
 \* Visits an MultiplyAssignExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

  - **Usage**  
 \* Visits a MultiplyExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

  - **Usage**  
 \* Visits a NotEqualExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.NotExpression node )
```

  - **Usage**  
 \* Visits a NotExpression
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
```

  - **Usage**  
 \* Visits a ObjectFieldAccess
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
```

  - **Usage**  
 \* Visits a ObjectMethodCall
  - **Parameters**  
 \* node - the node to visit

---
- *visit*  

```
public Object visit( koala.dynamicjava.tree.OrExpression node )
```

- **Usage**
    - \* Visits a OrExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
```

  - **Usage**
    - \* Visits an PackageDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PlusExpression node )
```

  - **Usage**
    - \* Visits a PlusExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostDecrement node )
```

  - **Usage**
    - \* Visits a PostDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement node )
```

  - **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.QualifiedName node )
      - Usage
        - \* Visits a QualifiedName
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ReferenceType node )
      - Usage
        - \* Visits a ReferenceType
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
      - Usage
        - \* Visits an RemainderAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderExpression node )
      - Usage
        - \* Visits a RemainderExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ReturnStatement node )
      - Usage
        - \* Visits a ReturnStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
      - Usage
        - \* Visits an ShiftLeftAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
      - Usage
        - \* Visits a ShiftLeftExpression
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )  
 – **Usage**  
   \* Visits an ShiftRightAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )  
 – **Usage**  
   \* Visits a ShiftRightExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAllocation node )  
 – **Usage**  
   \* Visits an SimpleAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )  
 – **Usage**  
   \* Visits an SimpleAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )  
 – **Usage**  
   \* Visits a StaticFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticMethodCall node )  
 – **Usage**  
   \* Visits a StaticMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )  
 – **Usage**  
   \* Visits an SubtractAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractExpression node )  


---

- **Usage**
    - \* Visits a SubtractExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )
```

  - **Usage**
    - \* Visits a SuperFieldAccess
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SuperMethodCall node )
```

  - **Usage**
    - \* Visits a SuperMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchBlock node )
```

  - **Usage**
    - \* Visits a SwitchBlock
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement node )
```

  - **Usage**
    - \* Visits a SwitchStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )
```

  - **Usage**
    - \* Visits a SynchronizedStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression node )
```

  - **Usage**
    - \* Visits a ThisExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement node )
```

  - **Usage**
    - \* Visits a ThrowStatement
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.TryStatement node )
      - Usage
        - \* Visits a TryStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.TypeExpression node )
      - Usage
        - \* Visits a TypeExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )
      - Usage
        - \* Visits an UnsignedShiftRightAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression node )
      - Usage
        - \* Visits a UnsignedShiftRightExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.VariableDeclaration node )
      - Usage
        - \* Visits a VariableDeclaration
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.WhileStatement node )
      - Usage
        - \* Visits a WhileStatement
      - Parameters
        - \* node - the node to visit

### 8.1.5 CLASS FileFinder

---

This class represents an object that manages a set of path where to find files.

DECLARATION

---

```
public class FileFinder
extends java.lang.Object
```

FIELDS

---

- private List paths
  - The paths

CONSTRUCTORS

---

- *FileFinder*  
 public **FileFinder**( )
  - **Usage**
    - \* Creates a new file finder

METHODS

---

- *addPath*  
 public void **addPath**( java.lang.String path )
  - **Usage**
    - \* Adds a search path. This path becomes the one with the highest priority.
  - **Parameters**
    - \* path - the path to add
- *findFile*  
 public File **findFile**( java.lang.String name )
  - **Usage**
    - \* Searches and returns a file.
  - **Parameters**
    - \* name - the base name of the file to find
  - **Exceptions**
    - \* java.io.IOException - if no file was found

### 8.1.6 CLASS ImportationManager

---

The instances of this class manages the importation clauses.

The **declarePackageImport** method imports a new package. This one has the highest priority over the imported packages when a lookup is made to find a class.

The **declareClassImport** method imports a new class. This one has the highest priority over the same suffix imported class.



DECLARATION

---

```
public class ImportationManager
extends java.lang.Object
implements java.lang.Cloneable
```

CONSTRUCTORS

---

• *ImportationManager*

```
public ImportationManager( java.lang.ClassLoader  cl )
```

– **Usage**

- \* Creates a new importation manager. The manager is initialized with two predefined import-on-demand clauses: "java.lang" (the java language package) and "" (the anonymous package)

– **Parameters**

- \* cl - the class loader to use

• *ImportationManager*

```
protected ImportationManager( koala.dynamicjava.util.ImportationManager
im )
```

– **Usage**

- \* Copy constructor

METHODS

---

• *clone*

```
public Object clone( )
```

– **Usage**

- \* Returns a copy of this object

• *declareClassImport*

```
public void declareClassImport( java.lang.String  cname )
```

– **Usage**

- \* Declares a new single-type-import clause

– **Parameters**

- \* cname - the fully qualified class name

– **Exceptions**

- \* java.lang.ClassNotFoundException - if the class cannot be found

• *declarePackageImport*

```
public void declarePackageImport( java.lang.String  pkg )
```

– **Usage**

- \* Declares a new import-on-demand clause

---

– **Parameters**

\* **pkg** - the package name

---

• *findInnerClass*

protected Class **findInnerClass**( java.lang.String **s** )

– **Usage**

\* Searches for an inner class from its name in the dotted notation

---

• *getCurrentPackage*

public String **getCurrentPackage**( )

– **Usage**

\* Returns the current package

---

• *getImportOnDemandClauses*

public List **getImportOnDemandClauses**( )

– **Usage**

\* Returns the import-on-demand clauses

---

• *getOuterNames*

protected List **getOuterNames**( java.lang.String **cname** )

– **Usage**

\* Returns a list of the outer classes names

---

• *getSingleTypeImportClauses*

public List **getSingleTypeImportClauses**( )

– **Usage**

\* Returns the single-type-import clauses

---

• *hasSuffix*

protected boolean **hasSuffix**( java.lang.String **c1**, java.lang.String **c2** )

– **Usage**

\* Tests whether the fully qualified class name c1 ends with c2

---

• *lookupClass*

public Class **lookupClass**( java.lang.String **cname**, java.lang.String **ccname** )

– **Usage**

\* Loads the class that match to the given name in the source file

– **Parameters**

\* **cname** - the name of the class to find

\* **ccname** - the name of the current class or null

– **Returns** - the class found

– **Exceptions**

\* java.lang.ClassNotFoundException - if the class cannot be loaded

---

- *setClassLoader*  
`public void setClassLoader( java.lang.ClassLoader cl )`
  - **Usage**
    - \* Sets the class loader

---
- *setCurrentPackage*  
`public void setCurrentPackage( java.lang.String pkg )`
  - **Usage**
    - \* Sets the current package. This has no influence on the behaviour of the `lookupClass` method.
  - **Parameters**
    - \* `pkg` - the package name

### 8.1.7 CLASS `LibraryFinder`

---

The instances of `LibraryFinder` are used to locate files with given suffixes.

#### DECLARATION

---

```
public class LibraryFinder
extends koala.dynamicjava.util.FileFinder
```

#### FIELDS

---

- private List suffixes
  - The file suffixes

#### CONSTRUCTORS

---

- *LibraryFinder*  
`public LibraryFinder( )`
  - **Usage**
    - \* Creates a new library finder
  - **Parameters**
    - \* `suffix` - the suffix of the source files

#### METHODS

---

- *addSuffix*  
`public void addSuffix( java.lang.String s )`
  - **Usage**

\* Adds a file suffix, for example ".java"

---

- *findCompilationUnit*

public File findCompilationUnit( java.lang.String cname )

- **Usage**

- \* Looks for the dynamic class with the given name

- **Parameters**

- \* **cname** - the fully qualified name of the class to find

- **Returns** - the file that contains the class

- **Exceptions**

- \* java.lang.ClassNotFoundException - if the class cannot be loaded

---

- *findCompilationUnitName*

public String findCompilationUnitName( java.lang.String cname )

- **Usage**

- \* Finds the path where the given class is possibly stored

- **Parameters**

- \* **cname** - the fully qualified name of the class to find

- **Returns** - the name of the root class

## METHODS INHERITED FROM CLASS koala.dynamicjava.util.FileFinder

---

( in 8.1.5, page 606)

- *addPath*

public void addPath( java.lang.String path )

- **Usage**

- \* Adds a search path. This path becomes the one with the highest priority.

- **Parameters**

- \* **path** - the path to add

---

- *findFile*

public File findFile( java.lang.String name )

- **Usage**

- \* Searches and returns a file.

- **Parameters**

- \* **name** - the base name of the file to find

- **Exceptions**

- \* java.io.IOException - if no file was found

### 8.1.8 CLASS LocalizedMessageReader

---

The instances of this class read localized messages in resource files. The messages in the file are templates. Context specific strings are inserted where '%n' patterns can be found. A '%' character is represented with '%%'.

DECLARATION

---

```
public class LocalizedMessageReader
extends java.lang.Object
```

FIELDS

---

- private static final char ESCAPE\_CHAR
  - The escape character
- private ResourceBundle bundle
  - The resource bundle

CONSTRUCTORS

---

- *LocalizedMessageReader*  
**public LocalizedMessageReader( java.lang.String name )**
  - **Usage**
    - \* Creates a new message reader
  - **Parameters**
    - \* **name** - the name of the resource

METHODS

---

- *getMessage*  
**public String getMessage( java.lang.String key, java.lang.String [] strings )**
  - **Usage**
    - \* Gets a message
  - **Parameters**
    - \* **key** - the message key
    - \* **strings** - the strings to insert in the message

### 8.1.9 CLASS ReflectionUtilities

---

This class contains a collection of utility methods for reflection.

DECLARATION

---

```
public class ReflectionUtilities
extends java.lang.Object
```

## CONSTRUCTORS

---

- *ReflectionUtilities*

**protected ReflectionUtilities( )**

- **Usage**

- \* This class contains only static methods, so it is not useful to create instances of it.

## METHODS

---

- *getConstructors*

**public static List getConstructors( java.lang.Class cl, int params )**

- **Usage**

- \* Gets all the constructors in the given class or super classes, even the redefined constructors are returned.

- **Parameters**

- \* **cl** - the class where the constructor was declared
- \* **params** - the number of parameters

- **Returns** - a list that contains the found constructors, an empty list if no matching constructor was found.

---

- *getField*

**public static Field getField( java.lang.Class cl, java.lang.String name )**

- **Usage**

- \* Returns a field with the given name declared in the given class or in the superclasses of the given class

- **Parameters**

- \* **cl** - the class where the field must look for the field
  - \* **name** - the name of the field
- 

- *getMethods*

**public static List getMethods( java.lang.Class cl, java.lang.String name, int params )**

- **Usage**

- \* Gets all the methods with the given name in the given class or super classes. Even the redefined methods are returned.

- **Parameters**

- \* **cl** - the class where the method was declared
- \* **name** - the name of the method
- \* **params** - the number of parameters

- **Returns** - a list that contains the found methods, an empty list if no matching method was found.

---

- *getOuterField*

**public static Field getOuterField( java.lang.Class cl, java.lang.String name )**

---

– **Usage**

- \* Returns a field with the given name declared in one of the outer classes of the given class

– **Parameters**

- \* **c1** - the inner class
  - \* **name** - the name of the field
- 

- *hasCompatibleSignatures*

```
public static boolean hasCompatibleSignatures( java.lang.Class [] a1,
java.lang.Class [] a2 )
```

– **Usage**

- \* For each element (class) of the given arrays, tests if the first array element is assignable from the second array element. The two arrays are assumed to have the same length.
- 

- *isCompatible*

```
public static boolean isCompatible( java.lang.Class c1, java.lang.Class c2
)
```

– **Usage**

- \* Whether 'c1' is assignable from 'c2'
- 

- *lookupConstructor*

```
public static Constructor lookupConstructor( java.lang.Class cl,
java.lang.Class [] ac )
```

– **Usage**

- \* Looks for a constructor in the given class or in super classes of this class.

– **Parameters**

- \* **c1** - the class of which the constructor is a member
  - \* **ac** - the arguments classes (possibly not the exact declaring classes)
- 

- *lookupMethod*

```
public static Method lookupMethod( java.lang.Class cl, java.lang.String
name, java.lang.Class [] ac )
```

– **Usage**

- \* Looks for a method in the given class or in super classes of this class.

– **Parameters**

- \* **c1** - the class of which the method is a member
  - \* **name** - the name of the method
  - \* **ac** - the arguments classes (possibly not the exact declaring classes)
- 

- *lookupMethod*

```
public static Method lookupMethod( java.lang.Class cl, java.lang.String
name, java.util.List ac )
```

– **Usage**

- \* Looks for a method in the given class or in super classes of this class.

– **Parameters**

- \* **c1** - the class of which the method is a member
  - \* **name** - the name of the method
  - \* **ac** - the arguments classes (possibly not the exact declaring classes)
- 

- *lookupOuterMethod*

```
public static Method lookupOuterMethod( java.lang.Class  cl,
java.lang.String  name, java.lang.Class [] ac )
```

- **Usage**

- \* Looks up for a method in an outer classes of this class.

- **Parameters**

- \* **c1** - the inner class
      - \* **name** - the name of the method
      - \* **ac** - the arguments classes (possibly not the exact declaring classes)
- 

- *selectTheMostSpecificConstructor*

```
protected static Constructor selectTheMostSpecificConstructor(
java.lang.reflect.Constructor  c1, java.lang.reflect.Constructor  c2 )
```

- **Usage**

- \* Returns the constructor with the most specific signature. It is assumed that m1 and m2 have the same number of parameters.
- 

- *selectTheMostSpecificMethod*

```
protected static Method selectTheMostSpecificMethod(
java.lang.reflect.Method  m1, java.lang.reflect.Method  m2 )
```

- **Usage**

- \* Returns the method with the most specific signature. It is assumed that m1 and m2 have the same number of parameters.



## Chapter 9

# Package `jeliot.mcode`

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Classes</b>	
<b>Code</b> .....	617
<i>This class contains the MCode language constructs as a final static variables that can be used in DynamicJava during extraction of information in MCode and interpretation of the MCode in MCode interpreter.</i>	
<b>Command</b> .....	622
<i>Currently this class is not used in Jeliot 3.</i>	
<b>Interpreter</b> .....	623
<i>The MCode interpreter that interprets the MCode received during the interpretation in DynamicJava.</i>	
<b>InterpreterError</b> .....	627
<i>This class encapsulates all the errors related to the interpretation of the user program.</i>	
<b>MCodeUtilities</b> .....	628
<i>This class contains helper methods for the MCode language extraction and interpretation.</i>	
<hr/>	

## 9.1 Classes

### 9.1.1 CLASS Code

---

This class contains the MCode language constructs as a final static variables that can be used in DynamicJava during extraction of information in MCode and interpretation of the MCode in MCode interpreter.

#### DECLARATION

---

<pre>public class Code <b>extends</b> java.lang.Object</pre>
--

#### FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle used in this package
- public static final String DELIM
  - Delimiter of the mcode expressions
- public static final String LOC\_DELIM
  - Delimiter of the mcode location expressions
- public static final String UNKNOWN
  - Symbol for an unknown value
- public static final int NO\_REFERENCE
  - Symbol for no reference
- public static final int NOT\_FINAL
  - Symbol for a non-final variable
- public static final int FINAL
  - Symbol for a final variable
- public static final String TRUE
  - Symbolic presentation for true value
- public static final String FALSE
  - Symbolic presentation for false value
- public static final String REFERENCE
  - Symbolic presentation for null reference
- public static final int LEFT
  - Left-hand side of the expression.

- public static final int RIGHT
  - Right-hand side of the expression.
- public static final int BEGIN
  - Beginning of the expression
- public static final int TO
  - Indicates that a value is going to be assigned to expression defined in referenced variable
- public static final int ERROR
  - Error statement
- public static final int END
  - End of Program
- public static final int CONSCN
  - Constructor call number to determine when the constructor call is finished. (SPECIAL) used for a hack
- public static final int A
  - Assignment
- public static final int AE
  - Add Expression
- public static final int SE
  - Subtract Expression
- public static final int ME
  - Multiplication Expression
- public static final int DE
  - Division Expression
- public static final int RE
  - Remainder Expression
- public static final int BITOR
  - Bitwise OR operator
- public static final int BITXOR
  - Bitwise XOR operator
- public static final int BITAND
  - Bitwise AND operator
- public static final int LSHIFT
  - Bitwise left shift operator

- public static final int RSHIFT
  - Bitwise right shift operator
- public static final int URSHIFT
  - Bitwise unsigned right shift operator
- public static final int PLUS
  - pression
- public static final int MINUS
  - pression
- public static final int PIE
  - Post Increment Expression
- public static final int PRIE
  - Pre Increment Expression
- public static final int PDE
  - Post Decrement Expression
- public static final int PRDE
  - Pre Decrement Expression
- public static final int COMP
  - Complement Expression
- public static final int XOR
  - Xor Expression
- public static final int AND
  - And Expression
- public static final int OR
  - Or Expression
- public static final int EE
  - Equal Expression
- public static final int NE
  - Not Equal Expression
- public static final int GT
  - Greater Than
- public static final int LE
  - Less Expression
- public static final int LQE
  - Less or Equal Expression

- public static final int GQT
  - Greater or Equal Expression
- public static final int NO
  - Boolean Not
- public static final int QN
  - Qualified Name
- public static final int L
  - Literal
- public static final int VD
  - Variable Declaration
- public static final int SCOPE
  - Opening and closing a scope
- public static final int IFT
  - If Then Statement
- public static final int IFTE
  - If Then Else Statement
- public static final int BR
  - Break statement
- public static final int WHI
  - While statement
- public static final int FOR
  - For statement
- public static final int CONT
  - Continue statement
- public static final int DO
  - Do-While statement
- public static final int SWIBF
  - Switch block found
- public static final int SWITCHB
  - Switch block begins
- public static final int SWITCH
  - Switch
- public static final int OUTPUT
  - Output statement

- public static final int INPUT
  - Input statement
- public static final int INPUTTED
  - Input statement
- public static final int SMC
  - Static Method Call
- public static final int P
  - Parameter
- public static final int PARAMETERS
  - Parameters list
- public static final int R
  - Return Statement
- public static final int SMCC
  - Static Method call closed
- public static final int MD
  - Method declaration
- public static final int OMC
  - Object method call
- public static final int OMCC
  - Object method call close
- public static final int SA
  - Simple class allocation
- public static final int SAC
  - Simple class allocation close
- public static final int OFA
  - Object field access
- public static final int AA
  - Array allocation
- public static final int AAC
  - Array access
- public static final int AL
  - Array length
- public static final int CLASS
  - Starting the information for a class.

- public static final int END\_CLASS
  - Ends the information for a class.
- public static final int CONSTRUCTOR
  - Indicates a constructor in the class information
- public static final int METHOD
  - Indicates a method in the class information
- public static final int FIELD
  - Indicates a field in the class information

## CONSTRUCTORS

---

- *Code*

```
private Code( )
```

  - **Usage**
    - \* This should be never used.

## 9.1.2 CLASS Command

---

Currently this class is not used in Jeliot 3.

## DECLARATION

---

```
public class Command
extends java.lang.Object
```

## FIELDS

---

- private int expressionReference
  -
- private int type
  -

## CONSTRUCTORS

---

- *Command*

```
protected Command( )
```
- *Command*

```
public Command( int t, int er )
```

  - **Parameters**
    - \* **t** -
    - \* **er** -

## METHODS

- *getExpressionReference*  
`public int getExpressionReference( )`  
 – Returns -

---

- *getType*  
`public int getType( )`  
 – Returns -

---

- *setExpressionReference*  
`public void setExpressionReference( int er )`  
 – Parameters  
   \* er -

---

- *setType*  
`public void setType( int t )`  
 – Parameters  
   \* t -

## 9.1.3 CLASS Interpreter

The MCode interpreter that interprets the MCode received during the interpretation in DynamicJava.

## DECLARATION

```
public class Interpreter
extends java.lang.Object
```

## FIELDS

- private Director director  
 –
- private BufferedReader ecode  
 –
- private PrintWriter input  
 –
- private String programCode  
 –



- private boolean running
  -
- private boolean start
  -
- private boolean firstLineRead
  -
- private boolean invokingMethod
  -
- private boolean returned
  - Keeps track of current return value
- private Value returnValue
  -
- private Actor returnActor
  -
- private long returnExpressionCounter
  -
- private Stack commands
  -
- private Stack exprs
  -
- private Hashtable values
  -
- private Hashtable variables
  -
- private Hashtable instances
  -
- private Stack methodInvocation
  -
- private Hashtable postIncsDecs
  -
- private ClassInfo currentClass
  -
- private Hashtable classes
  -

- private String line
  -
- private Object currentMethodInvocation
  - currentMethodInvocation keeps track of all the information that is collected during the method invocation. Cells: 0: Method name 1: Class/Object expression 2: Parameter values 3: Parameter types 4: Parameter names 5: Highlight info for invocation 6: Highlight info for declaration 7: Parameter expression references 8: Object reference if method is constructor or object method
- private Stack objectCreation
  -
- private boolean readNew
  -
- private boolean constructorCall
  - Related to Super method calls in constructor's first line in classes with inheritance.
- private Stack constructorCalls
  -
- private Vector superMethods
  -
- private Vector superMethodsReading
  -
- private long superMethodCallNumber
  -
- private static ResourceBundle bundle
  - The resource bundle for mcode messages
- private static ResourceBundle propertiesBundle
  - The resource bundle for mcode properties

## CONSTRUCTORS

---

- *Interpreter*  
protected Interpreter( )
- *Interpreter*  
 public **Interpreter**( java.io.BufferedReader **r**, jeliot.theater.Director **d**,  
 java.lang.String **programCode**, java.io.PrintWriter **pr** )
  - **Parameters**
    - \* **r** -
    - \* **d** -
    - \* **programCode** -
    - \* **pr** -

METHODS

---

• *checkInstancesForRemoval*

```
public void checkInstancesForRemoval( )
```

– **Usage**

```
* Not in use at the moment.
```

---

• *createNewInstance*

```
public ObjectFrame createNewInstance( jeliot.lang.ClassInfo ci,
jeliot.theater.Highlight h )
```

– **Parameters**

```
* ci -
```

```
* h -
```

– **Returns -**• *doPostIncDec*

```
public void doPostIncDec( java.lang.Object [] postIncDecInfo )
```

– **Parameters**

```
* postIncDecInfo -
```

---

• *emptyScratch*

```
public boolean emptyScratch( )
```

– **Returns -**• *execute*

```
public void execute( )
```

---

• *handleExpression*

```
private void handleExpression( jeliot.lang.Value val, long
expressionCounter )
```

– **Parameters**

```
* val -
```

```
* expressionCounter -
```

---

• *initialize*

```
public void initialize( )
```

– **Usage**

```
* Initializes the
```

---

• *interpret*

```
public void interpret( java.lang.String line )
```

– **Usage**

```
* Handles the interpretation of the single line of the mcode.
```

– **Parameters**

```
* line -
```

- 
- *readLine*  
`public String readLine( )`  
 – **Returns -**

---

  - *readNew*  
`public boolean readNew( )`  
 – **Returns -**

---

  - *removeInstances*  
`public void removeInstances( )`  
 – **Usage**  
   \* Not in use at the moment

---

  - *starting*  
`public boolean starting( )`  
 – **Returns -**

#### 9.1.4 CLASS InterpreterError

---

This class encapsulates all the errors related to the interpretation of the user program.

##### DECLARATION

---

```
public class InterpreterError
extends java.lang.Object
```

##### CONSTRUCTORS

---

- *InterpreterError*  
`public InterpreterError( java.lang.String message )`  
 – **Parameters**  
   \* `message` -

---

- *InterpreterError*  
`public InterpreterError( java.lang.String message, jeliot.theater.Highlight h )`  
 – **Usage**  
   \* Constructs an `InterpreterException` from an error message and a highlighting information.  
 – **Parameters**  
   \* `message` -  
   \* `h` -

## METHODS

- *getHighlight*  
`public Highlight getHighlight( )`  
 – **Returns** -

---

- *getMessage*  
`public String getMessage( )`  
 – **Usage**  
     \* Returns the detailed message  
 – **Returns** -

## 9.1.5 CLASS MCodeUtilities

This class contains helper methods for the MCode language extraction and interpretation.

## DECLARATION

```
public class MCodeUtilities
extends java.lang.Object
```

## FIELDS

- `public static final int COMP`  
 – Complement
- `public static final int MINUS`  
 –
- `public static final int MINUSMINUS`  
 – PreDecrement
- `public static final int NOT`  
 – Logical Not
- `public static final int PLUS`  
 –
- `public static final int PLUSPLUS`  
 – PreIncrement
- `public static final int POSTMINUSMINUS`  
 – PostDecrement
- `public static final int POSTPLUSPLUS`

- PostIncrement
- public static final int AND
  - Bitwise And
- public static final int ANDAND
  - Logical And
- public static final int DIV
  - Division
- public static final int EQEQ
  - Equality
- public static final int GT
  - Greater Than
- public static final int GTEQ
  - Greater Than Or Equal
- public static final int INSTANCEOF
  - Instance Of
- public static final int LSHIFT
  - Bitwise Left Shift
- public static final int LT
  - Lesser Than
- public static final int LTEQ
  - Lesser Than or Equal
- public static final int SUBTRACT
  - Substraction
- public static final int MOD
  - Remainder
- public static final int MULT
  - Multiplication
- public static final int NOTEQ
  - Not Equality
- public static final int OR
  - Bitwise Or
- public static final int OROR
  - Logical Or
- public static final int ADD

- Addition
- public static final int RSHIFT
  - Bitwise Right Shift
- public static final int URSHIFT
  - Bitwise Unsigned Right Shift
- public static final int XOR
  - Bitwise Xor
- public static final int LXOR
  - Logical Xor
- public static final int VOID
  - Void type
- public static final int BOOLEAN
  - Boolean type
- public static final int BYTE
  - Byte type
- public static final int SHORT
  - Short type
- public static final int INT
  - Int type
- public static final int LONG
  - Long type
- public static final int CHAR
  - Char type
- public static final int FLOAT
  - Float type
- public static final int DOUBLE
  - Double type
- public static final int STRING
  - String type
- public static final int REFERENCE
  - Referent type
- private static PrintWriter writer
  -
- private static BufferedReader reader

- 
- private static boolean redirectOutput
  - Hack flag to get the output into see below
- private static Vector redirectBuffer
  - Buffer to store the redirection orders from parameters collected in TreeInterpreter
- public static Stack redirectBufferStack
  - Stack with the redirect buffers
- public static Stack numParametersStack
  -
- private static int numParameters
  -

## CONSTRUCTORS

---

- *MCodeUtilities*  
 private **MCodeUtilities**( )
  - **Usage**
    - \* Should be never used.

## METHODS

---

- *addToRedirectBuffer*  
 public static void **addToRedirectBuffer**( java.lang.String string )
 

---
- *argToString*  
 public static String **argToString**( java.util.List argnames )
  - **Parameters**
    - \* argnames -
  - **Returns** -

---
- *arrayToString*  
 public static String **arrayToString**( java.lang.Object [] array )
  - **Parameters**
    - \* array -
  - **Returns** -

---
- *changeComponentTypeToPrintableForm*  
 public static String **changeComponentTypeToPrintableForm**(  
 java.lang.String type )
  - **Parameters**
    - \* type -



– **Returns -**

- *clearNumParameters*

public static void clearNumParameters( )

- *clearRedirectBuffer*

public static void clearRedirectBuffer( )

– **Parameters**

\* string -

- *findNumber*

public static int findNumber( java.lang.String from, java.lang.String identifier )

– **Parameters**

\* from -

\* identifier -

– **Returns -**

- *getDefaultValue*

public static String getDefaultValue( java.lang.String type )

– **Parameters**

\* type -

– **Returns -**

- *getHashCode*

public static String getHashCode( java.lang.String str )

– **Parameters**

\* str -

– **Returns -**

- *getNumberOfDimensions*

public static int getNumberOfDimensions( java.lang.String type )

– **Parameters**

\* type -

– **Returns -**

- *getNumParameters*

public static int getNumParameters( )

– **Returns -**

- *getRedirectBuffer*

public static Vector getRedirectBuffer( )

- *getRedirectOutput*

public static boolean getRedirectOutput( )

– **Returns -**

- *incNumParameters*  


---

```
public static void incNumParameters( )
```
- *isArray*  

```
public static boolean isArray( java.lang.String type )
```

  - **Parameters**
    - \* type -
  - **Returns** -

---
- *isBinary*  

```
public static boolean isBinary( int operator )
```

  - **Parameters**
    - \* operator -
  - **Returns** -

---
- *isPrimitive*  

```
public static boolean isPrimitive( java.lang.String type )
```

  - **Parameters**
    - \* type -
  - **Returns** -

---
- *isUnary*  

```
public static boolean isUnary( int operator )
```

  - **Parameters**
    - \* operator -
  - **Returns** -

---
- *makeHighlight*  

```
public static Highlight makeHighlight( java.lang.String h )
```

  - **Parameters**
    - \* h -
  - **Returns** -

---
- *parameterArrayToString*  

```
public static String parameterArrayToString( java.lang.Object [] array )
```

  - **Parameters**
    - \* array -
  - **Returns** -

---
- *readChar*  

```
public static Object readChar( )
```

  - **Returns** -

---
- *readDouble*  

```
public static Object readDouble( )
```

---

– **Returns** -

---

- *readInt*

public static Object readInt( )

– **Returns** -

---

- *readString*

public static Object readString( )

– **Returns** -

---

- *replace*

public static String replace( java.lang.String from, java.lang.String c,  
java.lang.String with )

– **Parameters**

\* from -

\* c -

\* with -

– **Returns** -

---

- *resolveBinOperator*

public static int resolveBinOperator( int operator )

– **Parameters**

\* operator -

– **Returns** -

---

- *resolveComponentType*

public static String resolveComponentType( java.lang.String type )

– **Parameters**

\* type -

– **Returns** -

---

- *resolveType*

public static int resolveType( java.lang.String type )

– **Parameters**

\* type -

– **Returns** -

---

- *resolveUnOperator*

public static int resolveUnOperator( int operator )

– **Parameters**

\* operator -

– **Returns** -

---

- *setReader*

public static void setReader( java.io.BufferedReader r )

- **Usage**

- \* For input handling.

- **Parameters**

- \* **r** -

---

- *setRedirectOutput*

```
public static void setRedirectOutput( boolean value )
```

- **Parameters**

- \* **boolean1** -

---

- *setWriter*

```
public static void setWriter( java.io.PrintWriter w )
```

- **Parameters**

- \* **w** -

---

- *write*

```
public static void write( java.lang.String str )
```

- **Parameters**

- \* **str** -

---

- *writeRedirectBuffer*

```
public static void writeRedirectBuffer( java.util.Vector redirectBuffer )
```

- **Returns** -

## Chapter 10

### Package

### koala.dynamicjava.interpreter.throwable

*Package Contents*

*Page*

---

#### Classes

<b>BreakException</b> .....	637
<i>Thrown to indicate that a break statement has been reached</i>	
<b>ContinueException</b> .....	639
<i>Thrown to indicate that a continue statement has been reached</i>	
<b>ReturnException</b> .....	641
<i>This error is thrown by an interpreted throw statement</i>	
<b>ThrownException</b> .....	644
<i>This error is thrown by an interpreted throw statement</i>	

---

## 10.1 Classes

### 10.1.1 CLASS **BreakException**

---

Thrown to indicate that a break statement has been reached

#### DECLARATION

---

```
public class BreakException
extends java.lang.RuntimeException
```

#### SERIALIZABLE FIELDS

---

- private String label
  - The label

#### FIELDS

---

- private String label
  - The label

#### CONSTRUCTORS

---

- *BreakException*  
**public BreakException( java.lang.String m )**
  - **Usage**
    - \* Constructs an **BreakException** with the specified detail message.
  - **Parameters**
    - \* **m** - the detail message.
- *BreakException*  
**public BreakException( java.lang.String m, java.lang.String l )**
  - **Usage**
    - \* Constructs an **BreakException** with the specified detail message and label.
  - **Parameters**
    - \* **m** - the detail message.
    - \* **l** - the label

METHODS

---

- *getLabel*  
 public String **getLabel**( )  
 – **Usage**  
 \* Returns the label of the statement that thrown the exception

---

- *isLabeled*  
 public boolean **isLabeled**( )  
 – **Usage**  
 \* Tests whether the statement was labeled

METHODS INHERITED FROM CLASS java.lang.RuntimeException

---

METHODS INHERITED FROM CLASS java.lang.Exception

---

METHODS INHERITED FROM CLASS java.lang.Throwable

---

- *fillInStackTrace*  
 public synchronized native Throwable **fillInStackTrace**( )
- *getCause*  
 public Throwable **getCause**( )
- *getLocalizedMessage*  
 public String **getLocalizedMessage**( )
- *getMessage*  
 public String **getMessage**( )
- *getOurStackTrace*  
 private synchronized StackTraceElement **getOurStackTrace**( )
- *getStackTrace*  
 public StackTraceElement **getStackTrace**( )
- *getStackTraceDepth*  
 private native int **getStackTraceDepth**( )
- *getStackTraceElement*  
 private native StackTraceElement **getStackTraceElement**( int    )
- *initCause*  
 public synchronized Throwable **initCause**( java.lang.Throwable    )
- *printStackTrace*  
 public void **printStackTrace**( )
- *printStackTrace*  
 public void **printStackTrace**( java.io.PrintStream    )
- *printStackTrace*  
 public void **printStackTrace**( java.io.PrintWriter    )

- *printStackTraceAsCause*  
 private void **printStackTraceAsCause**( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
 private void **printStackTraceAsCause**( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
 public void **setStackTrace**( java.lang.StackTraceElement [] )
- *toString*  
 public String **toString**( )
- *writeObject*  
 private synchronized void **writeObject**( java.io.ObjectOutputStream )

### 10.1.2 CLASS ContinueException

---

Thrown to indicate that a continue statement has been reached

#### DECLARATION

---

```
public class ContinueException
extends java.lang.RuntimeException
```

#### SERIALIZABLE FIELDS

---

- private String label
  - The label

#### FIELDS

---

- private String label
  - The label

#### CONSTRUCTORS

---

- *ContinueException*  
 public **ContinueException**( java.lang.String m )
  - **Usage**
    - \* Constructs an **ContinueException** with the specified detail message.
  - **Parameters**
    - \* m - the detail message.
- *ContinueException*  
 public **ContinueException**( java.lang.String m, java.lang.String l )



- **Usage**
  - \* Constructs an `ContinueException` with the specified detail message and label.
- **Parameters**
  - \* `m` - the detail message.
  - \* `l` - the label

## METHODS

---

- *getLabel*  
public String **getLabel**( )
  - **Usage**
    - \* Returns the label of the statement that thrown the exception
- *isLabeled*  
public boolean **isLabeled**( )
  - **Usage**
    - \* Tests whether the statement was labeled

## METHODS INHERITED FROM CLASS `java.lang.RuntimeException`

---

## METHODS INHERITED FROM CLASS `java.lang.Exception`

---

## METHODS INHERITED FROM CLASS `java.lang.Throwable`

---

- *fillInStackTrace*  
public synchronized native Throwable **fillInStackTrace**( )
- *getCause*  
public Throwable **getCause**( )
- *getLocalizedMessage*  
public String **getLocalizedMessage**( )
- *getMessage*  
public String **getMessage**( )
- *getOurStackTrace*  
private synchronized StackTraceElement **getOurStackTrace**( )
- *getStackTrace*  
public StackTraceElement **getStackTrace**( )
- *getStackTraceDepth*  
private native int **getStackTraceDepth**( )
- *getStackTraceElement*  
private native StackTraceElement **getStackTraceElement**( int    )
- *initCause*  
public synchronized Throwable **initCause**( java.lang.Throwable    )

- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

### 10.1.3 CLASS ReturnException

---

This error is thrown by an interpreted throw statement

#### DECLARATION

---

```
public class ReturnException
extends koala.dynamicjava.interpreter.error.ExecutionError
```

#### SERIALIZABLE FIELDS

---

- private boolean withValue
  - Whether the return has a value
- private Object value
  - The returned object

#### FIELDS

---

- private boolean withValue
  - Whether the return has a value
- private Object value
  - The returned object

CONSTRUCTORS

---

• *ReturnException*

```
public ReturnException( java.lang.String s, koala.dynamicjava.tree.Node n
)
```

## – Usage

\* Constructs an *ReturnException* with a value

---

• *ReturnException*

```
public ReturnException( java.lang.String s, java.lang.Object o,
koala.dynamicjava.tree.Node n )
```

## – Usage

\* Constructs an *ReturnExceptionError* with the specified detail message, filename, line, column and exception.

## – Parameters

\* *e* - the return exception

\* *n* - the node in the syntax tree where the error occurs

METHODS

---

• *getValue*

```
public Object getValue( )
```

## – Usage

\* Returns the value returned

---

• *hasValue*

```
public boolean hasValue( )
```

## – Usage

\* Whether or not the return statement had a value

METHODS INHERITED FROM CLASS *koala.dynamicjava.interpreter.error.ExecutionError*

---

( in 17.1.2, page 1332)

• *getMessage*

```
public String getMessage( )
```

## – Usage

\* Returns the error message string of this exception

---

• *getNode*

```
public Node getNode( )
```

## – Usage

\* Returns the syntax tree node where the error occurs

---

• *printStackTrace*

```
public void printStackTrace( )
```

- **Usage**
    - \* Overridden to delegate to `printStackTrace(PrintStream)` to print nested exception information.
  - **See Also**
    - \* `koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintStream)`
- 
- *printStackTrace*  
**public void printStackTrace( java.io.PrintStream s )**
    - **Usage**
      - \* Overridden to delegate to `printStackTrace(PrintWriter)` to print nested exception information.
    - **See Also**
      - \* `koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintWriter)`
- 
- *printStackTrace*  
**public void printStackTrace( java.io.PrintWriter w )**
    - **Usage**
      - \* Handles all calls to `printStackTrace()`, printing the stack trace of the current exception, and also that of its cause.

#### METHODS INHERITED FROM CLASS `java.lang.Error`

---

#### METHODS INHERITED FROM CLASS `java.lang.Throwable`

---

- *fillInStackTrace*  
**public synchronized native Throwable fillInStackTrace( )**
- *getCause*  
**public Throwable getCause( )**
- *getLocalizedMessage*  
**public String getLocalizedMessage( )**
- *getMessage*  
**public String getMessage( )**
- *getOurStackTrace*  
**private synchronized StackTraceElement getOurStackTrace( )**
- *getStackTrace*  
**public StackTraceElement getStackTrace( )**
- *getStackTraceDepth*  
**private native int getStackTraceDepth( )**
- *getStackTraceElement*  
**private native StackTraceElement getStackTraceElement( int )**
- *initCause*  
**public synchronized Throwable initCause( java.lang.Throwable )**
- *printStackTrace*  
**public void printStackTrace( )**
- *printStackTrace*  
**public void printStackTrace( java.io.PrintStream )**

- *printStackTrace*  
`public void printStackTrace( java.io.PrintWriter )`
- *printStackTraceAsCause*  
`private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )`
- *printStackTraceAsCause*  
`private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )`
- *setStackTrace*  
`public void setStackTrace( java.lang.StackTraceElement [] )`
- *toString*  
`public String toString( )`
- *writeObject*  
`private synchronized void writeObject( java.io.ObjectOutputStream )`

#### 10.1.4 CLASS *ThrownException*

---

This error is thrown by an interpreted throw statement

##### DECLARATION

---

```
public class ThrownException
extends koala.dynamicjava.interpreter.error.ExecutionError
```

##### CONSTRUCTORS

---

- *ThrownException*  
`public ThrownException( java.lang.Throwable e )`  
  - **Usage**  
\* Constructs an *ThrownExceptionError* with no detail message.
- *ThrownException*  
`public ThrownException( java.lang.Throwable e,  
koala.dynamicjava.tree.Node n )`  
  - **Usage**  
\* Constructs an *ThrownExceptionError* with the specified detail message, filename, line, column and exception.
  - **Parameters**  
\* **e** - the thrown exception  
\* **n** - the node in the syntax tree where the error occurs

##### METHODS

---

- *getException*  
`public Throwable getException( )`  
  - **Usage**  
\* Returns the exception that causes this error throwing

METHODS INHERITED FROM CLASS `koala.dynamicjava.interpreter.error.ExecutionError`

---

( in 17.1.2, page 1332)

- *getMessage*  
`public String getMessage( )`  
  - **Usage**  
 \* Returns the error message string of this exception

---

- *getNode*  
`public Node getNode( )`  
  - **Usage**  
 \* Returns the syntax tree node where the error occurs

---

- *printStackTrace*  
`public void printStackTrace( )`  
  - **Usage**  
 \* Overridden to delegate to `printStackTrace(PrintStream)` to print nested exception information.
  - **See Also**  
 \*  
`koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintStream)`

---

- *printStackTrace*  
`public void printStackTrace( java.io.PrintStream s )`  
  - **Usage**  
 \* Overridden to delegate to `printStackTrace(PrintWriter)` to print nested exception information.
  - **See Also**  
 \*  
`koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintWriter)`

---

- *printStackTrace*  
`public void printStackTrace( java.io.PrintWriter w )`  
  - **Usage**  
 \* Handles all calls to `printStackTrace()`, printing the stack trace of the current exception, and also that of its cause.

METHODS INHERITED FROM CLASS `java.lang.Error`

---

METHODS INHERITED FROM CLASS `java.lang.Throwable`

---

- *fillInStackTrace*  
`public synchronized native Throwable fillInStackTrace( )`
- *getCause*  
`public Throwable getCause( )`
- *getLocalizedMessage*  
`public String getLocalizedMessage( )`

- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

## Chapter 11

# Package

## koala.dynamicjava.interpreter.context

Package Contents

Page

### Interfaces

<b>Context</b> .....	649
<i>The classes that implements this interface represent contexts of execution</i>	
<b>SimpleContext</b> .....	655
<i>The classes that implements this interface represent contexts of execution</i>	

### Classes

<b>GlobalContext</b> .....	658
<i>A global context.</i>	
<b>GlobalContext.CompilationUnitVisitor</b> .....	668
<i>To find a class in a compilation unit</i>	
<b>GlobalContext.CompilationUnitVisitor.MembersVisitor</b> .....	683
<i>To find a class in a compilation unit</i>	
<b>GlobalContext.PseudoClassLoader</b> .....	696
<i>To test the existance of a class without loading it</i>	
<b>GlobalContext.PseudoError</b> .....	699
<i>To test the existance of a class without loading it</i>	
<b>MethodContext</b> .....	701
<i>A method method context.</i>	
<b>MethodModificationError</b> .....	714
<i>This error is thrown by a context when it modify the syntax tree</i>	
<b>NoSuchFunctionException</b> .....	716
<i>Thrown when a particular method cannot be found.</i>	
<b>StaticContext</b> .....	718
<i>A static method context.</i>	
<b>VariableContext</b> .....	730
<i>This class encapsulates the behaviour of Java scopes.</i>	
<b>VariableContext.AbstractVariable</b> .....	733
<i>To store the variables</i>	
<b>VariableContext.Constant</b> .....	734
<i>To store the constants</i>	
<b>VariableContext.Link</b> .....	735
<i>To store one scope</i>	



<b>VariableContext.LinkFactory</b> .....	736
<i>To manage the creation of scopes and links</i>	
<b>VariableContext.Scope</b> .....	736
<i>A table which maps a string with an object</i>	
<b>VariableContext.Scope.Entry</b> .....	738
<i>To manage collisions</i>	
<b>VariableContext.Scope.EntryFactory</b> .....	738
<i>To create an entry</i>	
<b>VariableContext.Variable</b> .....	739
<i>To store the variables</i>	

---

## 11.1 Interfaces

### 11.1.1 INTERFACE Context

The classes that implements this interface represent contexts of execution

#### DECLARATION

```
public interface Context
implements SimpleContext
```

#### METHODS

- *classExists*  
 public boolean **classExists**( java.lang.String name )  
 – **Usage**  
   \* Whether a simple identifier is a class  
 – **Parameters**  
   \* name - the identifier

---

- *createName*  
 public Expression **createName**( koala.dynamicjava.tree.Node node, koala.dynamicjava.tree.IdentifierToken name )  
 – **Usage**  
   \* Creates the tree that is associated with the given name  
 – **Parameters**  
   \* node - the current node  
   \* name - the variable name  
 – **Exceptions**  
   \* java.lang.IllegalStateException - if the variable is not defined

---

- *declareClassImport*  
 public void **declareClassImport**( java.lang.String cname )  
 – **Usage**  
   \* Declares a new single-type-import clause  
 – **Parameters**  
   \* cname - the fully qualified class name  
 – **Exceptions**  
   \* java.lang.ClassNotFoundException - if the class cannot be found

---

- *declarePackageImport*  
 public void **declarePackageImport**( java.lang.String pkg )  
 – **Usage**  
   \* Declares a new import-on-demand clause

- **Parameters**
    - \* **pkg** - the package name

---
- *defineClass*

```
public void defineClass( koala.dynamicjava.tree.TypeDeclaration  node )
```

    - **Usage**
      - \* Defines a class from its syntax tree
    - **Parameters**
      - \* **node** - the class declaration

---
  - *defineFunction*

```
public void defineFunction( koala.dynamicjava.tree.MethodDeclaration  node )
```

      - **Usage**
        - \* Defines a MethodDeclaration as a function
      - **Parameters**
        - \* **node** - the function declaration

---
    - *exists*

```
public boolean exists( java.lang.String  name )
```

        - **Usage**
          - \* Whether a simple identifier represents an existing variable or field in this context
        - **Parameters**
          - \* **name** - the identifier

---
      - *getAccessible*

```
public boolean getAccessible( )
```

          - **Usage**
            - \* Returns the accessibility state of this context.

---
        - *getCurrentPackage*

```
public String getCurrentPackage( )
```

            - **Usage**
              - \* Returns the current package

---
          - *getDefaultQualifier*

```
public Node getDefaultQualifier( koala.dynamicjava.tree.Node  node )
```

              - **Usage**
                - \* Returns the default qualifier for this context
              - **Parameters**
                - \* **node** - the current node

---
            - *getDefaultQualifier*

```
public Node getDefaultQualifier( koala.dynamicjava.tree.Node  node,
java.lang.String  tname )
```

- **Usage**
    - \* Returns the default qualifier for this context
  - **Parameters**
    - \* **node** - the current node
    - \* **tname** - the qualifier of 'this'
- 

- *getField*

```
public Field getField( java.lang.Class fc, java.lang.String fn )
```

- **Usage**
    - \* Looks for a field
  - **Parameters**
    - \* **fc** - the field class
    - \* **fn** - the field name
  - **Exceptions**
    - \* `java.lang.NoSuchFieldException` - if the field cannot be found
    - \* `koala.dynamicjava.util.AmbiguousFieldException` - if the field is ambiguous
- 

- *getFunctions*

```
public List getFunctions( )
```

- **Usage**
    - \* Returns the defined functions
- 

- *getHiddenArgument*

```
public Object getHiddenArgument( )
```

- **Usage**
    - \* Returns the default argument to pass to methods in this context
- 

- *getImportationManager*

```
public ImportationManager getImportationManager( )
```

- **Usage**
    - \* Returns the importation manager
- 

- *getInterpreter*

```
public Interpreter getInterpreter( )
```

- **Usage**
    - \* Returns the current interpreter
- 

- *getModifier*

```
public LeftHandSideModifier getModifier(
koala.dynamicjava.tree.ObjectFieldAccess node )
```

- **Usage**
    - \* Returns the modifier that match the given node
  - **Parameters**
    - \* **node** - a tree node
-

- *getModifier*

```
public LeftHandSideModifier getModifier(
    koala.dynamicjava.tree.QualifiedName node )
```

- **Usage**

- \* Returns the modifier that match the given node

- **Parameters**

- \* **node** - a tree node

---

- *getModifier*

```
public LeftHandSideModifier getModifier(
    koala.dynamicjava.tree.StaticFieldAccess node )
```

- **Usage**

- \* Returns the modifier that match the given node

- **Parameters**

- \* **node** - a tree node

---

- *getModifier*

```
public LeftHandSideModifier getModifier(
    koala.dynamicjava.tree.SuperFieldAccess node )
```

- **Usage**

- \* Returns the modifier that match the given node

- **Parameters**

- \* **node** - a tree node

---

- *getSuperField*

```
public Field getSuperField( koala.dynamicjava.tree.Node node,
    java.lang.String fn )
```

- **Usage**

- \* Looks for a field in the super class

- **Parameters**

- \* **node** - the current node

- \* **fn** - the field name

- **Exceptions**

- \* `java.lang.NoSuchFieldException` - if the field cannot be found

- \* `koala.dynamicjava.util.AmbiguousFieldException` - if the field is ambiguous

---

- *invokeConstructor*

```
public Object invokeConstructor( koala.dynamicjava.tree.ClassAllocation
    node, java.lang.Object [] args )
```

- **Usage**

- \* Invokes a constructor

- **Parameters**

- \* **node** - the ClassAllocation node

- \* **args** - the arguments

---

- *invokeConstructor*

```
public Object invokeConstructor( koala.dynamicjava.tree.SimpleAllocation
node, java.lang.Object [] args )
```

- **Usage**

- \* Invokes a constructor

- **Parameters**

- \* **node** - the SimpleAllocation node

- \* **args** - the arguments

---

- *isDefined*

```
public boolean isDefined( java.lang.String name )
```

- **Usage**

- \* Tests whether a variable or a field is defined in this context

- **Parameters**

- \* **name** - the name of the entry

- **Returns** - false if the variable is undefined

---

- *lookupClass*

```
public Class lookupClass( java.lang.String cname )
```

- **Usage**

- \* Looks for a class

- **Parameters**

- \* **cname** - the class name

- **Exceptions**

- \* `java.lang.ClassNotFoundException` - if the class cannot be found

---

- *lookupClass*

```
public Class lookupClass( java.lang.String cname, java.lang.String
ccname )
```

- **Usage**

- \* Looks for a class (context-free lookup)

- **Parameters**

- \* **cname** - the class name

- \* **ccname** - the fully qualified name of the context class

- **Exceptions**

- \* `java.lang.ClassNotFoundException` - if the class cannot be found

---

- *lookupConstructor*

```
public Constructor lookupConstructor( java.lang.Class c, java.lang.Class
[] params )
```

- **Usage**

- \* Looks for a constructor

- **Parameters**

- \* **c** - the class of the constructor

- \* **params** - the parameter types

- **Exceptions**
    - \* `java.lang.NoSuchMethodException` - if the constructor cannot be found

---
- *lookupFunction*

```
public MethodDeclaration lookupFunction( java.lang.String  mname,
java.lang.Class []  params )
```

  - **Usage**
    - \* Looks for a function
  - **Parameters**
    - \* `mname` - the function name
    - \* `params` - the parameter types
  - **Exceptions**
    - \* `koala.dynamicjava.interpreter.context.NoSuchFunctionException` - if the function cannot be found

---
- *lookupMethod*

```
public Method lookupMethod( koala.dynamicjava.tree.Node  prefix,
java.lang.String  mname, java.lang.Class []  params )
```

  - **Usage**
    - \* Looks for a method
  - **Parameters**
    - \* `prefix` - the method prefix
    - \* `mname` - the method name
    - \* `params` - the parameter types
  - **Exceptions**
    - \* `java.lang.NoSuchMethodException` - if the method cannot be found

---
- *lookupSuperMethod*

```
public Method lookupSuperMethod( koala.dynamicjava.tree.Node  node,
java.lang.String  mname, java.lang.Class []  params )
```

  - **Usage**
    - \* Looks for a super method
  - **Parameters**
    - \* `node` - the current node
    - \* `mname` - the method name
    - \* `params` - the parameter types
  - **Exceptions**
    - \* `java.lang.NoSuchMethodException` - if the method cannot be found

---
- *setAccessible*

```
public void setAccessible( boolean  accessible )
```

  - **Usage**
    - \* Allows the scripts to access private fields.

---
- *setAdditionalClassLoaderContainer*

```
public void setAdditionalClassLoaderContainer(
koala.dynamicjava.interpreter.ClassLoaderContainer  clc )
```

- **Usage**
    - \* Sets the additional class loader container
- 
- *setCurrentPackage*

```
public void setCurrentPackage( java.lang.String  pkg )
```

    - **Usage**
      - \* Sets the current package
    - **Parameters**
      - \* **pkg** - the package name
- 
- *setFunctions*

```
public void setFunctions( java.util.List  l )
```

    - **Usage**
      - \* Sets the defined functions
- 
- *setImportationManager*

```
public void setImportationManager(
koala.dynamicjava.util.ImportationManager  im )
```

    - **Usage**
      - \* Sets the importation manager
- 
- *setPropertyies*

```
public Class setProperties( koala.dynamicjava.tree.ClassAllocation  node,
java.lang.Class  c, java.lang.Class [] args, java.util.List  memb )
```

    - **Usage**
      - \* Sets the properties of a ClassAllocation node
    - **Parameters**
      - \* **node** - the allocation node
      - \* **c** - the class of the constructor
      - \* **args** - the classes of the arguments of the constructor
      - \* **memb** - the class members
- 
- *setPropertyies*

```
public Class setProperties( koala.dynamicjava.tree.SimpleAllocation  node,
java.lang.Class  c, java.lang.Class [] args )
```

    - **Usage**
      - \* Sets the properties of a SimpleAllocation node
    - **Parameters**
      - \* **node** - the allocation node
      - \* **c** - the class of the constructor
      - \* **args** - the classes of the arguments of the constructor

### 11.1.2 INTERFACE SimpleContext

---

The classes that implements this interface represent contexts of execution



DECLARATION

---

public interface SimpleContext
--------------------------------

METHODS

---

- *define*  
 public void **define**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
   \* Defines a new variable in the current scope  
 – **Parameters**  
   \* **name** - the name of the new entry  
   \* **value** - the value of the entry  
 – **Exceptions**  
   \* java.lang.IllegalStateException - if the variable is already defined

---

- *defineConstant*  
 public void **defineConstant**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
   \* Defines a new constant variable in the current scope  
 – **Parameters**  
   \* **name** - the name of the new entry  
   \* **value** - the value of the entry  
 – **Exceptions**  
   \* java.lang.IllegalStateException - if the variable is already defined

---

- *defineVariables*  
 public void **defineVariables**( java.util.Set **vars** )  
 – **Usage**  
   \* Defines the given variables

---

- *enterScope*  
 public void **enterScope**( )  
 – **Usage**  
   \* Enters a scope

---

- *enterScope*  
 public void **enterScope**( java.util.Set **entries** )  
 – **Usage**  
   \* Enters a scope and defines the given entries to null.  
 – **Parameters**  
   \* **entries** - a set of string

---

- *get*  
 public Object **get**( java.lang.String name )  
 – **Usage**  
   \* Returns the value of a variable with the given name  
 – **Parameters**  
   \* name - the name of the value to get  
 – **Exceptions**  
   \* java.lang.IllegalStateException - if the variable is not defined  


---
- *getConstants*  
 public Map **getConstants**( )  
 – **Usage**  
   \* Creates a map that contains the constants in this context  


---
- *getCurrentScopeVariableNames*  
 public Set **getCurrentScopeVariableNames**( )  
 – **Usage**  
   \* Returns the current scope variables (strings) in a set  


---
- *getCurrentScopeVariables*  
 public Set **getCurrentScopeVariables**( )  
 – **Usage**  
   \* Returns the current scope variables (strings) in a set  


---
- *isDefinedVariable*  
 public boolean **isDefinedVariable**( java.lang.String name )  
 – **Usage**  
   \* Tests whether a variable is defined in this context  
 – **Parameters**  
   \* name - the name of the entry  
 – **Returns** - false if the variable is undefined  


---
- *isFinal*  
 public boolean **isFinal**( java.lang.String name )  
 – **Usage**  
   \* Tests whether a variable is final in this context  
 – **Parameters**  
   \* name - the name of the entry  
 – **Returns** - false if the variable is not final  
 – **Exceptions**  
   \* java.lang.IllegalStateException - if the variable is not defined  


---
- *leaveScope*  
 public Set **leaveScope**( )

- **Usage**
    - \* Leaves the current scope
  - **Returns** - the set of the variables (strings) defined in the current scope
- 
- *set*

```
public void set( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of a defined variable
    - **Parameters**
      - \* **name** - the name of the new entry
      - \* **value** - the value of the entry
    - **Exceptions**
      - \* `java.lang.IllegalStateException` - if the variable is not defined
- 
- *setConstant*

```
public void setConstant( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Defines a new constant variable in the current scope
    - **Parameters**
      - \* **name** - the name of the new entry
      - \* **value** - the value of the entry
- 
- *setVariable*

```
public void setVariable( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Defines a new variable in the current scope
    - **Parameters**
      - \* **name** - the name of the new entry
      - \* **value** - the value of the entry

## 11.2 Classes

### 11.2.1 CLASS GlobalContext

---

A global context.

DECLARATION

---

```
public class GlobalContext
extends koala.dynamicjava.interpreter.context.VariableContext
implements Context
```

CONSTRUCTORS

---

• *GlobalContext*

```
public GlobalContext( koala.dynamicjava.interpreter.Interpreter i )
```

– **Usage**

\* Creates a new context

– **Parameters**

\* *i* - the interpreter

---

• *GlobalContext*

```
public GlobalContext( koala.dynamicjava.interpreter.Interpreter i,  
java.lang.ClassLoader cl )
```

– **Usage**

\* Creates a new context

– **Parameters**

\* *i* - the interpreter

\* *cl* - the classloader to use

\* *cl2* - the additional classloader

---

• *GlobalContext*

```
public GlobalContext( koala.dynamicjava.interpreter.Interpreter i,  
java.util.Set entries )
```

– **Usage**

\* Creates a new context initialized with the given entries defined in the initial scope.

– **Parameters**

\* *i* - the interpreter

\* *entries* - a set of string

METHODS

---

• *classExists*

```
public boolean classExists( java.lang.String name )
```

– **Usage**

\* Whether a simple identifier is a class

– **Parameters**

\* *name* - the identifier

---

• *createClassArrayInitializer*

```
protected ArrayInitializer createClassArrayInitializer( )
```

– **Usage**

\* Creates an initializer for the variable class array used to implement inner classes

---

• *createName*

```
public Expression createName( koala.dynamicjava.tree.Node node,  
koala.dynamicjava.tree.IdentifierToken name )
```

- **Usage**
    - \* Creates the tree that is associated with the given name
  - **Parameters**
    - \* **node** - the current node
    - \* **name** - the variable name
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if the variable is not defined
- 
- *declareClassImport*  
`public void declareClassImport( java.lang.String cname )`
    - **Usage**
      - \* Declares a new single-type-import clause
    - **Parameters**
      - \* **cname** - the fully qualified class name
    - **Exceptions**
      - \* `java.lang.ClassNotFoundException` - if the class cannot be found
- 
- *declarePackageImport*  
`public void declarePackageImport( java.lang.String pkg )`
    - **Usage**
      - \* Declares a new import-on-demand clause
    - **Parameters**
      - \* **pkg** - the package name
- 
- *defineClass*  
`public void defineClass( koala.dynamicjava.tree.TypeDeclaration node )`
    - **Usage**
      - \* Defines a class from its syntax tree
    - **Parameters**
      - \* **node** - the class declaration
- 
- *defineFunction*  
`public void defineFunction( koala.dynamicjava.tree.MethodDeclaration node )`
    - **Usage**
      - \* Defines a MethodDeclaration as a function
    - **Parameters**
      - \* **node** - the function declaration
- 
- *exists*  
`public boolean exists( java.lang.String name )`
    - **Usage**
      - \* Whether a simple identifier represents an existing variable or field or type in this context.
    - **Parameters**
-

- 
- \* **name** - the identifier
- 
- *getAccessible*  
 public boolean **getAccessible**( )  
 – **Usage**  
 \* Returns the accessibility state of this context.
- 
- *getAdditionalClassLoader*  
 protected ClassLoader **getAdditionalClassLoader**( )  
 – **Usage**  
 \* Gets the additional class loader
- 
- *getCurrentPackage*  
 public String **getCurrentPackage**( )  
 – **Usage**  
 \* Returns the current package
- 
- *getDefaultQualifier*  
 public Node **getDefaultQualifier**( koala.dynamicjava.tree.Node **node** )  
 – **Usage**  
 \* Returns the default qualifier for this context  
 – **Parameters**  
 \* **node** - the current node
- 
- *getDefaultQualifier*  
 public Node **getDefaultQualifier**( koala.dynamicjava.tree.Node **node**,  
 java.lang.String **tname** )  
 – **Usage**  
 \* Returns the default qualifier for this context  
 – **Parameters**  
 \* **node** - the current node  
 \* **tname** - the qualifier of 'this'
- 
- *getField*  
 public Field **getField**( java.lang.Class **fc**, java.lang.String **fn** )  
 – **Usage**  
 \* Looks for a field  
 – **Parameters**  
 \* **fc** - the field class  
 \* **fn** - the field name  
 – **Exceptions**  
 \* java.lang.NoSuchFieldException - if the field cannot be find  
 \* koala.dynamicjava.util.AmbiguousFieldException - if the field is ambiguous
- 
- *getFunctions*  
 public List **getFunctions**( )

- **Usage**
    - \* Returns the defined functions

---
- *getHiddenArgument*

```
public Object getHiddenArgument( )
```

    - **Usage**
      - \* Returns the default argument to pass to methods in this context

---
  - *getImportationManager*

```
public ImportationManager getImportationManager( )
```

    - **Usage**
      - \* Returns the importation manager

---
  - *getInterpreter*

```
public Interpreter getInterpreter( )
```

    - **Usage**
      - \* Returns the current interpreter

---
  - *getModifier*

```
public LeftHandSideModifier getModifier(
koala.dynamicjava.tree.ObjectFieldAccess node )
```

    - **Usage**
      - \* Returns the modifier that match the given node
    - **Parameters**
      - \* **node** - a tree node

---
  - *getModifier*

```
public LeftHandSideModifier getModifier(
koala.dynamicjava.tree.QualifiedName node )
```

    - **Usage**
      - \* Returns the modifier that match the given node
    - **Parameters**
      - \* **node** - a tree node

---
  - *getModifier*

```
public LeftHandSideModifier getModifier(
koala.dynamicjava.tree.StaticFieldAccess node )
```

    - **Usage**
      - \* Returns the modifier that match the given node
    - **Parameters**
      - \* **node** - a tree node

---
  - *getModifier*

```
public LeftHandSideModifier getModifier(
koala.dynamicjava.tree.SuperFieldAccess node )
```

- **Usage**
    - \* Returns the modifier that match the given node
  - **Parameters**
    - \* **node** - a tree node
- 
- *getPackageName*  
 protected String **getPackageName**( java.lang.Class **c** )
    - **Usage**
      - \* Gets the package name for the given class
- 
- *getSuperField*  
 public Field **getSuperField**( koala.dynamicjava.tree.Node **node**,  
 java.lang.String **fn** )
    - **Usage**
      - \* Looks for a field in the super class
    - **Parameters**
      - \* **node** - the current node
      - \* **fn** - the field name
    - **Exceptions**
      - \* java.lang.NoSuchFieldException - if the field cannot be find
      - \* koala.dynamicjava.util.AmbiguousFieldException - if the field is ambiguous
- 
- *invokeConstructor*  
 public Object **invokeConstructor**( koala.dynamicjava.tree.ClassAllocation  
 node, java.lang.Object [] **args** )
    - **Usage**
      - \* Invokes a constructor
    - **Parameters**
      - \* **node** - the ClassAllocation node
      - \* **args** - the arguments
- 
- *invokeConstructor*  
 public Object **invokeConstructor**( koala.dynamicjava.tree.SimpleAllocation  
 node, java.lang.Object [] **args** )
    - **Usage**
      - \* Invokes a constructor
    - **Parameters**
      - \* **node** - the SimpleAllocation node
      - \* **args** - the arguments
- 
- *isDefined*  
 public boolean **isDefined**( java.lang.String **name** )
    - **Usage**
      - \* Tests whether a variable is defined in this context
    - **Parameters**
      - \* **name** - the name of the entry



- **Returns** - false if the variable is undefined

---

- *lookupClass*

```
public Class lookupClass( java.lang.String  cname )
```

- **Usage**
  - \* Looks for a class
- **Parameters**
  - \* `cname` - the class name
- **Exceptions**
  - \* `java.lang.ClassNotFoundException` - if the class cannot be found

---

- *lookupClass*

```
public Class lookupClass( java.lang.String  cname, java.lang.String
cname )
```

- **Usage**
  - \* Looks for a class (context-free lookup)
- **Parameters**
  - \* `cname` - the class name
  - \* `cname` - the fully qualified name of the context class
- **Exceptions**
  - \* `java.lang.ClassNotFoundException` - if the class cannot be found

---

- *lookupConstructor*

```
public Constructor lookupConstructor( java.lang.Class  c, java.lang.Class
[] params )
```

- **Usage**
  - \* Looks for a constructor
- **Parameters**
  - \* `c` - the class of the constructor
  - \* `params` - the parameter types
- **Exceptions**
  - \* `java.lang.NoSuchMethodException` - if the constructor cannot be found

---

- *lookupFunction*

```
public MethodDeclaration lookupFunction( java.lang.String  mname,
java.lang.Class [] params )
```

- **Usage**
    - \* Looks for a function
  - **Parameters**
    - \* `mname` - the function name
    - \* `params` - the parameter types
  - **Exceptions**
    - \* `koala.dynamicjava.interpreter.context.NoSuchFunctionException` - if the function cannot be found
-

- *lookupMethod*

```
public Method lookupMethod( koala.dynamicjava.tree.Node prefix,
    java.lang.String mname, java.lang.Class [] params )
```

- **Usage**

- \* Looks for a method

- **Parameters**

- \* **prefix** - the method prefix
  - \* **mname** - the method name
  - \* **params** - the parameter types

- **Exceptions**

- \* `java.lang.NoSuchMethodException` - if the method cannot be found

---

- *lookupSuperMethod*

```
public Method lookupSuperMethod( koala.dynamicjava.tree.Node node,
    java.lang.String mname, java.lang.Class [] params )
```

- **Usage**

- \* Looks for a super method

- **Parameters**

- \* **node** - the current node
  - \* **mname** - the method name
  - \* **params** - the parameter types

- **Exceptions**

- \* `java.lang.NoSuchMethodException` - if the method cannot be find

---

- *setAccessFlag*

```
protected void setAccessFlag( java.lang.reflect.Member m )
```

- **Usage**

- \* Sets the access flag of a member

---

- *setAccessible*

```
public void setAccessible( boolean accessible )
```

- **Usage**

- \* Allows the scripts to access private fields.

---

- *setAdditionalClassLoaderContainer*

```
public void setAdditionalClassLoaderContainer(
    koala.dynamicjava.interpreter.ClassLoaderContainer clc )
```

- **Usage**

- \* Sets the additional class loader container

---

- *setCurrentPackage*

```
public void setCurrentPackage( java.lang.String pkg )
```

- **Usage**

- \* Sets the current package

- **Parameters**

- \* **pkg** - the package name

---

- *setFunctions*

```
public void setFunctions( java.util.List l )
```

- **Usage**

- \* Sets the defined functions

---

- *setImportationManager*

```
public void setImportationManager(
koala.dynamicjava.util.ImportationManager im )
```

- **Usage**

- \* Sets the importation manager

---

- *setPropertyies*

```
public Class setProperties( koala.dynamicjava.tree.ClassAllocation node,
java.lang.Class c, java.lang.Class [] args, java.util.List memb )
```

- **Usage**

- \* Sets the properties of a ClassAllocation node

- **Parameters**

- \* **node** - the allocation node
  - \* **c** - the class of the constructor
  - \* **args** - the classes of the arguments of the constructor
  - \* **memb** - the class members

---

- *setPropertyies*

```
public Class setProperties( koala.dynamicjava.tree.SimpleAllocation node,
java.lang.Class c, java.lang.Class [] cargs )
```

- **Usage**

- \* Sets the properties of a SimpleAllocation node

- **Parameters**

- \* **node** - the allocation node
  - \* **c** - the class of the constructor
  - \* **cargs** - the classes of the arguments of the constructor

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.context.VariableContext

---

( in 11.2.10, page 730)

- *define*

```
public void define( java.lang.String name, java.lang.Object value )
```

- **Usage**

- \* Defines a new variable in the current scope

- **Parameters**

- \* **name** - the name of the new entry
  - \* **value** - the value of the entry

- **Exceptions**

- \* `java.lang.IllegalStateException` - if the variable is already defined

---

- *defineConstant*  
 public void **defineConstant**( java.lang.String name, java.lang.Object value )  


---

  - **Usage**  
 \* Defines a new constant variable in the current scope
  - **Parameters**  
 \* name - the name of the new entry  
 \* value - the value of the entry
  - **Exceptions**  
 \* java.lang.IllegalStateException - if the variable is already defined

---
- *defineVariables*  
 public void **defineVariables**( java.util.Set vars )  


---

  - **Usage**  
 \* Defines the given variables

---
- *enterScope*  
 public void **enterScope**( )  


---

  - **Usage**  
 \* Enters a scope

---
- *enterScope*  
 public void **enterScope**( java.util.Set entries )  


---

  - **Usage**  
 \* Enters a scope and defines the given entries to null.
  - **Parameters**  
 \* entries - a set of string

---
- *get*  
 public Object **get**( java.lang.String name )  


---

  - **Usage**  
 \* Returns the value of a variable with the given name
  - **Parameters**  
 \* name - the name of the value to get
  - **Exceptions**  
 \* java.lang.IllegalStateException - if the variable is not defined

---
- *getConstants*  
 public Map **getConstants**( )  


---

  - **Usage**  
 \* Creates a map that contains the constants in this context

---
- *getCurrentScopeVariableNames*  
 public Set **getCurrentScopeVariableNames**( )  


---

  - **Usage**  
 \* Returns the current scope variables (strings) in a set

---
- *getCurrentScopeVariables*  
 public Set **getCurrentScopeVariables**( )  


---

  - **Usage**  
 \* Returns the current scope variables in a set

---
- *isDefinedVariable*  
 public boolean **isDefinedVariable**( java.lang.String name )

- **Usage**
    - \* Tests whether an entry is defined in this context
  - **Parameters**
    - \* **name** - the name of the entry
- 
- *isFinal*

```
public boolean isFinal( java.lang.String name )
```

    - **Usage**
      - \* Tests whether a variable is final in this context
    - **Parameters**
      - \* **name** - the name of the entry
    - **Returns** - false if the variable is not final
    - **Exceptions**
      - \* `java.lang.IllegalStateException` - if the variable is not defined
- 
- *leaveScope*

```
public Set leaveScope( )
```

    - **Usage**
      - \* Leaves the current scope
    - **Returns** - the set of variable defined in this scope
- 
- *set*

```
public void set( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a defined variable
    - **Parameters**
      - \* **name** - the name of the entry
      - \* **value** - the value of the entry
    - **Exceptions**
      - \* `java.lang.IllegalStateException` - if the variable is not defined or is final
- 
- *setConstant*

```
public void setConstant( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a constant variable in the current scope
    - **Parameters**
      - \* **name** - the name of the entry
      - \* **value** - the value of the entry
- 
- *setVariable*

```
public void setVariable( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a variable in the current scope
    - **Parameters**
      - \* **name** - the name of the entry
      - \* **value** - the value of the entry

## 11.2.2 CLASS GlobalContext.CompilationUnitVisitor

---

To find a class in a compilation unit

DECLARATION

---

```
private class GlobalContext.CompilationUnitVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

FIELDS

---

- private String className
  - The class to find
- private String currentPackage
  - The current package
- private ImportationManager importationManager
  - The importation manager
- private TreeClassLoader classLoader
  - The current class loader

CONSTRUCTORS

---

- *GlobalContext.CompilationUnitVisitor*  
**public GlobalContext.CompilationUnitVisitor( java.lang.String cname )**
  - **Usage**
    - \* Creates a new visitor
- *GlobalContext.CompilationUnitVisitor*  
**public GlobalContext.CompilationUnitVisitor( java.lang.String cname, koala.dynamicjava.util.ImportationManager im )**
  - **Usage**
    - \* Creates a new visitor

METHODS

---

- *visit*  
**public Object visit( koala.dynamicjava.tree.ClassDeclaration node )**
  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* **node** - the node to visit
- *visit*  
**public Object visit( koala.dynamicjava.tree.ImportDeclaration node )**

- **Usage**
    - \* Visits an ImportDeclaration
  - **Parameters**
    - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )
```

    - **Usage**
      - \* Visits an InterfaceDeclaration
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
```

    - **Usage**
      - \* Visits a PackageDeclaration
    - **Parameters**
      - \* **node** - the node to visit
    - **Returns** - null
- 
- *visitType*

```
private Object visitType( koala.dynamicjava.tree.TypeDeclaration node )
```

    - **Usage**
      - \* visits a TypeDeclaration

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.visitor.VisitorObject`

---

( in 5.2.1, page 160)

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

    - **Usage**
      - \* Visits an AddAssignExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

    - **Usage**
      - \* Visits a AddExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

    - **Usage**
      - \* Visits a AndExpression

- **Parameters**
  - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.ArrayAccess node )`
  - **Usage**
    - \* Visits a `ArrayAccess`
    - **Parameters**
    - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.ArrayAllocation node )`
  - **Usage**
    - \* Visits an `ArrayAllocation`
    - **Parameters**
    - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.ArrayInitializer node )`
  - **Usage**
    - \* Visits an `ArrayInitializer`
    - **Parameters**
    - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.ArrayType node )`
  - **Usage**
    - \* Visits a `ArrayType`
    - **Parameters**
    - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )`
  - **Usage**
    - \* Visits a `BitAndAssignExpression`
    - **Parameters**
    - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.BitAndExpression node )`
  - **Usage**
    - \* Visits a `BitAndExpression`
    - **Parameters**
    - \* `node` - the node to visit

---
- *visit*
  - `public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )`
  - **Usage**
    - \* Visits a `BitOrAssignExpression`
    - **Parameters**
    - \* `node` - the node to visit

---



- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )  
 – **Usage**  
   \* Visits a BitOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )  
 – **Usage**  
   \* Visits a BlockStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BreakStatement node )  
 – **Usage**  
   \* Visits a BreakStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.CastExpression node )  
 – **Usage**  
   \* Visits a CastExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.CatchStatement node )  
 – **Usage**  
   \* Visits a CatchStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassAllocation node )  
 – **Usage**  
   \* Visits an ClassAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassDeclaration node )  
 – **Usage**  
   \* Visits a ClassDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassInitializer node )

- **Usage**
    - \* Visits a ClassInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression node )
```

  - **Usage**
    - \* Visits a ComplementExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
```

  - **Usage**
    - \* Visits a ConditionalExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )
```

  - **Usage**
    - \* Visits a ConstructorDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )
```

  - **Usage**
    - \* Visits a ConstructorInvocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ContinueStatement node )
```

  - **Usage**
    - \* Visits a ContinueStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
```

  - **Usage**
    - \* Visits an DivideAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideExpression node )
```

  - **Usage**
    - \* Visits a DivideExpression
  - **Parameters**

- 
- \* node - the node to visit

---

• *visit*

public Object visit( koala.dynamicjava.tree.DoStatement node )

    - Usage
    - \* Visits a DoStatement
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.EmptyStatement node )

    - Usage
    - \* Visits an EmptyStatement
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.EqualExpression node )

    - Usage
    - \* Visits a EqualExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )

    - Usage
    - \* Visits a ExclusiveOrAssignExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )

    - Usage
    - \* Visits a ExclusiveOrExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.FieldDeclaration node )

    - Usage
    - \* Visits a FieldDeclaration
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.FormalParameter node )

    - Usage
    - \* Visits a FormalParameter
    - Parameters
    - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ForStatement node )  
 – **Usage**  
   \* Visits a ForStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FunctionCall node )  
 – **Usage**  
   \* Visits a FunctionCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterExpression node )  
 – **Usage**  
   \* Visits a GreaterExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )  
 – **Usage**  
   \* Visits a GreaterOrEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )  
 – **Usage**  
   \* Visits a IfThenElseStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenStatement node )  
 – **Usage**  
   \* Visits a IfThenStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ImportDeclaration node )  
 – **Usage**  
   \* Visits an ImportDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.InnerAllocation node )

- **Usage**
    - \* Visits an InnerAllocation
  - **Parameters**
    - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )
```

    - **Usage**
      - \* Visits an InnerClassAllocation
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceInitializer node )
```

    - **Usage**
      - \* Visits a InstanceInitializer
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
```

    - **Usage**
      - \* Visits an InstanceOfExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )
```

    - **Usage**
      - \* Visits a InterfaceDeclaration
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.LabeledStatement node )
```

    - **Usage**
      - \* Visits a LabeledStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.LessExpression node )
```

    - **Usage**
      - \* Visits a LessExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
```

    - **Usage**
      - \* Visits a LessOrEqualExpression
    - **Parameters**

- 
- \* node - the node to visit
  - - *visit*
    - public Object visit( koala.dynamicjava.tree.Literal node )
    - Usage
      - \* Visits a Literal
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
    - Usage
      - \* Visits a MethodDeclaration
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.MinusExpression node )
    - Usage
      - \* Visits a MinusExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
    - Usage
      - \* Visits an MultiplyAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
    - Usage
      - \* Visits a MultiplyExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
    - Usage
      - \* Visits a NotEqualExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.NotExpression node )
    - Usage
      - \* Visits a NotExpression
      - Parameters
      - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )  
 – **Usage**  
   \* Visits a ObjectFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )  
 – **Usage**  
   \* Visits a ObjectMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.OrExpression node )  
 – **Usage**  
   \* Visits a OrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PackageDeclaration node )  
 – **Usage**  
   \* Visits an PackageDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PlusExpression node )  
 – **Usage**  
   \* Visits a PlusExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PostDecrement node )  
 – **Usage**  
   \* Visits a PostDecrement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PostIncrement node )  
 – **Usage**  
   \* Visits a PostIncrement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PreDecrement node )  


---

- **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.QualifiedName node )
```

  - **Usage**
    - \* Visits a QualifiedName
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReferenceType node )
```

  - **Usage**
    - \* Visits a ReferenceType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
```

  - **Usage**
    - \* Visits an RemainderAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderExpression node )
```

  - **Usage**
    - \* Visits a RemainderExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReturnStatement node )
```

  - **Usage**
    - \* Visits a ReturnStatement
  - **Parameters**



- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
      - Usage
        - \* Visits an ShiftLeftAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
      - Usage
        - \* Visits a ShiftLeftExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
      - Usage
        - \* Visits an ShiftRightAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
      - Usage
        - \* Visits a ShiftRightExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
      - Usage
        - \* Visits an SimpleAllocation
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )
      - Usage
        - \* Visits an SimpleAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )
      - Usage
        - \* Visits a StaticFieldAccess
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticMethodCall node )  
 – **Usage**  
   \* Visits a StaticMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )  
 – **Usage**  
   \* Visits an SubtractAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractExpression node )  
 – **Usage**  
   \* Visits a SubtractExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )  
 – **Usage**  
   \* Visits a SuperFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperMethodCall node )  
 – **Usage**  
   \* Visits a SuperMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SwitchBlock node )  
 – **Usage**  
   \* Visits a SwitchBlock  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SwitchStatement node )  
 – **Usage**  
   \* Visits a SwitchStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )

- **Usage**
    - \* Visits a SynchronizedStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression node )
```

  - **Usage**
    - \* Visits a ThisExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement node )
```

  - **Usage**
    - \* Visits a ThrowStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement node )
```

  - **Usage**
    - \* Visits a TryStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression node )
```

  - **Usage**
    - \* Visits a TypeExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )
```

  - **Usage**
    - \* Visits an UnsignedShiftRightAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression node )
```

  - **Usage**
    - \* Visits a UnsignedShiftRightExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration node )
```

  - **Usage**
    - \* Visits a VariableDeclaration

- **Parameters**
  - \* **node** - the node to visit
- *visit*  

```
public Object visit( koala.dynamicjava.tree.WhileStatement  node )
```
- **Usage**
  - \* Visits a WhileStatement
- **Parameters**
  - \* **node** - the node to visit

### 11.2.3 CLASS GlobalContext.CompilationUnitVisitor.MembersVisitor

---

To find a class in a compilation unit

#### DECLARATION

---

```
private class GlobalContext.CompilationUnitVisitor.MembersVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

#### FIELDS

---

- private String outerName
  - The outer class

#### CONSTRUCTORS

---

- *GlobalContext.CompilationUnitVisitor.MembersVisitor*  

```
public GlobalContext.CompilationUnitVisitor.MembersVisitor(
    java.lang.String  cname )
```
- **Usage**
  - \* Creates a new visitor

#### METHODS

---

- *visit*  

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration  node )
```
  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* **node** - the node to visit
- 
- *visit*  

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration  node )
```

- **Usage**
    - \* Visits an InterfaceDeclaration
  - **Parameters**
    - \* **node** - the node to visit
- 
- *visitType*  
`private Object visitType( koala.dynamicjava.tree.TypeDeclaration node )`
    - **Usage**
      - \* visits a TypeDeclaration

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.visitor.VisitorObject`

---

( in 5.2.1, page 160)

- *visit*  
`public Object visit( koala.dynamicjava.tree.AddAssignExpression node )`
    - **Usage**
      - \* Visits an AddAssignExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*  
`public Object visit( koala.dynamicjava.tree.AddExpression node )`
    - **Usage**
      - \* Visits a AddExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*  
`public Object visit( koala.dynamicjava.tree.AndExpression node )`
    - **Usage**
      - \* Visits a AndExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayAccess node )`
    - **Usage**
      - \* Visits a ArrayAccess
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayAllocation node )`
    - **Usage**
      - \* Visits an ArrayAllocation
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*  
`public Object visit( koala.dynamicjava.tree.ArrayInitializer node )`

- **Usage**
    - \* Visits an ArrayInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayType  node )
```

  - **Usage**
    - \* Visits a ArrayType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndAssignExpression  node )
```

  - **Usage**
    - \* Visits a BitAndAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndExpression  node )
```

  - **Usage**
    - \* Visits a BitAndExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrAssignExpression  node )
```

  - **Usage**
    - \* Visits a BitOrAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitOrExpression  node )
```

  - **Usage**
    - \* Visits a BitOrExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BlockStatement  node )
```

  - **Usage**
    - \* Visits a BlockStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BreakStatement  node )
```

  - **Usage**
    - \* Visits a BreakStatement
  - **Parameters**

- 
- \* node - the node to visit

---

• *visit*

public Object visit( koala.dynamicjava.tree.CastExpression node )

    - Usage
    - \* Visits a CastExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.CatchStatement node )

    - Usage
    - \* Visits a CatchStatement
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ClassAllocation node )

    - Usage
    - \* Visits an ClassAllocation
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ClassDeclaration node )

    - Usage
    - \* Visits a ClassDeclaration
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ClassInitializer node )

    - Usage
    - \* Visits a ClassInitializer
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ComplementExpression node )

    - Usage
    - \* Visits a ComplementExpression
    - Parameters
    - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ConditionalExpression node )

    - Usage
    - \* Visits a ConditionalExpression
    - Parameters
    - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )  
 – **Usage**  
   \* Visits a ConstructorDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )  
 – **Usage**  
   \* Visits a ConstructorInvocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ContinueStatement node )  
 – **Usage**  
   \* Visits a ContinueStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )  
 – **Usage**  
   \* Visits an DivideAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideExpression node )  
 – **Usage**  
   \* Visits a DivideExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.DoStatement node )  
 – **Usage**  
   \* Visits a DoStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.EmptyStatement node )  
 – **Usage**  
   \* Visits an EmptyStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.EqualExpression node )



- **Usage**
    - \* Visits a EqualExpression
  - **Parameters**
    - \* `node` - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
    - **Usage**
      - \* Visits a ExclusiveOrAssignExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
    - **Usage**
      - \* Visits a ExclusiveOrExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.FieldDeclaration node )
    - **Usage**
      - \* Visits a FieldDeclaration
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.FormalParameter node )
    - **Usage**
      - \* Visits a FormalParameter
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.ForStatement node )
    - **Usage**
      - \* Visits a ForStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.FunctionCall node )
    - **Usage**
      - \* Visits a FunctionCall
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*  
public Object visit( koala.dynamicjava.tree.GreaterExpression node )
    - **Usage**
      - \* Visits a GreaterExpression
    - **Parameters**

- 
- \* node - the node to visit

---

• *visit*

public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )

    - Usage
      - \* Visits a GreaterOrEqualExpression
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )

    - Usage
      - \* Visits a IfThenElseStatement
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.IfThenStatement node )

    - Usage
      - \* Visits a IfThenStatement
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ImportDeclaration node )

    - Usage
      - \* Visits an ImportDeclaration
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.InnerAllocation node )

    - Usage
      - \* Visits an InnerAllocation
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )

    - Usage
      - \* Visits an InnerClassAllocation
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.InstanceInitializer node )

    - Usage
      - \* Visits a InstanceInitializer
    - Parameters
      - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )  
 – **Usage**  
   \* Visits an InstanceOfExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )  
 – **Usage**  
   \* Visits a InterfaceDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.LabeledStatement node )  
 – **Usage**  
   \* Visits a LabeledStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )  
 – **Usage**  
   \* Visits a LessExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )  
 – **Usage**  
   \* Visits a LessOrEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.Literal node )  
 – **Usage**  
   \* Visits a Literal  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.MethodDeclaration node )  
 – **Usage**  
   \* Visits a MethodDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.MinusExpression node )

- **Usage**
    - \* Visits a MinusExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

  - **Usage**
    - \* Visits an MultiplyAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

  - **Usage**
    - \* Visits a MultiplyExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

  - **Usage**
    - \* Visits a NotEqualExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotExpression node )
```

  - **Usage**
    - \* Visits a NotExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
```

  - **Usage**
    - \* Visits a ObjectFieldAccess
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
```

  - **Usage**
    - \* Visits a ObjectMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.OrExpression node )
```

  - **Usage**
    - \* Visits a OrExpression
  - **Parameters**

- - \* node - the node to visit
- *visit*  
public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
  - Usage
    - \* Visits an PackageDeclaration
  - Parameters
    - \* node - the node to visit
- *visit*  
public Object visit( koala.dynamicjava.tree.PlusExpression node )
  - Usage
    - \* Visits a PlusExpression
  - Parameters
    - \* node - the node to visit
- *visit*  
public Object visit( koala.dynamicjava.tree.PostDecrement node )
  - Usage
    - \* Visits a PostDecrement
  - Parameters
    - \* node - the node to visit
- *visit*  
public Object visit( koala.dynamicjava.tree.PostIncrement node )
  - Usage
    - \* Visits a PostIncrement
  - Parameters
    - \* node - the node to visit
- *visit*  
public Object visit( koala.dynamicjava.tree.PreDecrement node )
  - Usage
    - \* Visits a PreDecrement
  - Parameters
    - \* node - the node to visit
- *visit*  
public Object visit( koala.dynamicjava.tree.PreIncrement node )
  - Usage
    - \* Visits a PreIncrement
  - Parameters
    - \* node - the node to visit
- *visit*  
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
  - Usage
    - \* Visits a PrimitiveType
  - Parameters
    - \* node - the node to visit

- *visit*  
 public Object visit( koala.dynamicjava.tree.QualifiedName node )  
 – **Usage**  
   \* Visits a QualifiedName  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ReferenceType node )  
 – **Usage**  
   \* Visits a ReferenceType  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )  
 – **Usage**  
   \* Visits an RemainderAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderExpression node )  
 – **Usage**  
   \* Visits a RemainderExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ReturnStatement node )  
 – **Usage**  
   \* Visits a ReturnStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )  
 – **Usage**  
   \* Visits an ShiftLeftAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )  
 – **Usage**  
   \* Visits a ShiftLeftExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )

- **Usage**
    - \* Visits an ShiftRightAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
  - **Usage**
    - \* Visits a ShiftRightExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
  - **Usage**
    - \* Visits an SimpleAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )
  - **Usage**
    - \* Visits an SimpleAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )
  - **Usage**
    - \* Visits a StaticFieldAccess
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.StaticMethodCall node )
  - **Usage**
    - \* Visits a StaticMethodCall
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
  - **Usage**
    - \* Visits an SubtractAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*  
public Object visit( koala.dynamicjava.tree.SubtractExpression node )
  - **Usage**
    - \* Visits a SubtractExpression
  - **Parameters**

- 
- \* node - the node to visit

---

• *visit*

public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )

    - Usage
      - \* Visits a SuperFieldAccess
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.SuperMethodCall node )

    - Usage
      - \* Visits a SuperMethodCall
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.SwitchBlock node )

    - Usage
      - \* Visits a SwitchBlock
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.SwitchStatement node )

    - Usage
      - \* Visits a SwitchStatement
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )

    - Usage
      - \* Visits a SynchronizedStatement
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ThisExpression node )

    - Usage
      - \* Visits a ThisExpression
    - Parameters
      - \* node - the node to visit

---
  - *visit*

public Object visit( koala.dynamicjava.tree.ThrowStatement node )

    - Usage
      - \* Visits a ThrowStatement
    - Parameters
      - \* node - the node to visit

---



- *visit*  
 public Object visit( koala.dynamicjava.tree.TryStatement node )  
 – Usage  
   \* Visits a TryStatement  
 – Parameters  
   \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.TypeExpression node )  
 – Usage  
   \* Visits a TypeExpression  
 – Parameters  
   \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )  
 – Usage  
   \* Visits an UnsignedShiftRightAssignExpression  
 – Parameters  
   \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression node )  
 – Usage  
   \* Visits a UnsignedShiftRightExpression  
 – Parameters  
   \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.VariableDeclaration node )  
 – Usage  
   \* Visits a VariableDeclaration  
 – Parameters  
   \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.WhileStatement node )  
 – Usage  
   \* Visits a WhileStatement  
 – Parameters  
   \* node - the node to visit

#### 11.2.4 CLASS GlobalContext.PseudoClassLoader

---

To test the existence of a class without loading it

DECLARATION

---

```
protected class GlobalContext.PseudoClassLoader
extends java.lang.ClassLoader
```

CONSTRUCTORS

---

- *GlobalContext.PseudoClassLoader*  
protected **GlobalContext.PseudoClassLoader**( )

METHODS

---

- *findClass*  
protected **Class findClass**( java.lang.String name )
  - **Usage**
    - \* Finds the specified class.
  - **Parameters**
    - \* name - the name of the class
  - **Returns** - the resulting **Class** object
  - **Exceptions**
    - \* java.lang.ClassNotFoundException - if the class could not be find

METHODS INHERITED FROM CLASS java.lang.ClassLoader

---

- *<clinit>*  
static void <clinit>( )
- *addClass*  
void addClass( java.lang.Class )
- *check*  
private void check( )
- *checkCerts*  
private synchronized void checkCerts( java.lang.String , java.security.CodeSource )
- *checkPackageAccess*  
private void checkPackageAccess( java.lang.Class , java.security.ProtectionDomain )
- *clearAssertionStatus*  
public synchronized void clearAssertionStatus( )
- *compareCerts*  
private boolean compareCerts( java.security.cert.Certificate [] , java.security.cert.Certificate [] )
- *defineClass*  
protected final Class defineClass( byte [] , int , int )
- *defineClass*  
protected final Class defineClass( java.lang.String , byte [] , int , int )
- *defineClass*  
protected final Class defineClass( java.lang.String , byte [] , int , int , java.security.ProtectionDomain )
- *defineClass0*  
private native Class defineClass0( java.lang.String , byte [] , int , int , java.security.ProtectionDomain )

- *definePackage*  
protected Package definePackage( java.lang.String , java.lang.String ,  
java.lang.String , java.lang.String , java.lang.String , java.lang.String ,  
java.lang.String , java.net.URL )
- *desiredAssertionStatus*  
synchronized boolean desiredAssertionStatus( java.lang.String )
- *findBootstrapClass*  
private native Class findBootstrapClass( java.lang.String )
- *findBootstrapClass0*  
private Class findBootstrapClass0( java.lang.String )
- *findClass*  
protected Class findClass( java.lang.String )
- *findLibrary*  
protected String findLibrary( java.lang.String )
- *findLoadedClass*  
protected final native Class findLoadedClass( java.lang.String )
- *findNative*  
static long findNative( java.lang.ClassLoader , java.lang.String )
- *findResource*  
protected URL findResource( java.lang.String )
- *findResources*  
protected Enumeration findResources( java.lang.String )
- *findSystemClass*  
protected final Class findSystemClass( java.lang.String )
- *getBootstrapClassPath*  
static URLClassPath getBootstrapClassPath( )
- *getBootstrapResource*  
private static URL getBootstrapResource( java.lang.String )
- *getBootstrapResources*  
private static Enumeration getBootstrapResources( java.lang.String )
- *getCallerClassLoader*  
static ClassLoader getCallerClassLoader( )
- *getDefaultDomain*  
private synchronized ProtectionDomain getDefaultDomain( )
- *getPackage*  
protected Package getPackage( java.lang.String )
- *getPackages*  
protected Package getPackages( )
- *getParent*  
public final ClassLoader getParent( )
- *getResource*  
public URL getResource( java.lang.String )
- *getResourceAsStream*  
public InputStream getResourceAsStream( java.lang.String )
- *getResources*  
public final Enumeration getResources( java.lang.String )
- *getSystemClassLoader*  
public static ClassLoader getSystemClassLoader( )
- *getSystemResource*  
public static URL getSystemResource( java.lang.String )

- *getSystemResourceAsStream*  
public static InputStream **getSystemResourceAsStream**( java.lang.String    )
- *getSystemResources*  
public static Enumeration **getSystemResources**( java.lang.String    )
- *initializeJavaAssertionMaps*  
private void **initializeJavaAssertionMaps**( )
- *initializePath*  
private static String **initializePath**( java.lang.String    )
- *initSystemClassLoader*  
private static synchronized void **initSystemClassLoader**( )
- *isAncestor*  
boolean **isAncestor**( java.lang.ClassLoader    )
- *loadClass*  
public Class **loadClass**( java.lang.String    )
- *loadClass*  
protected synchronized Class **loadClass**( java.lang.String    , boolean    )
- *loadClassInternal*  
private synchronized Class **loadClassInternal**( java.lang.String    )
- *loadLibrary*  
static void **loadLibrary**( java.lang.Class    , java.lang.String    , boolean    )
- *loadLibrary0*  
private static boolean **loadLibrary0**( java.lang.Class    , java.io.File    )
- *registerNatives*  
private static native void **registerNatives**( )
- *resolveClass*  
protected final void **resolveClass**( java.lang.Class    )
- *resolveClass0*  
private native void **resolveClass0**( java.lang.Class    )
- *retrieveDirectives*  
private static native AssertionStatusDirectives **retrieveDirectives**( )
- *setClassAssertionStatus*  
public synchronized void **setClassAssertionStatus**( java.lang.String    , boolean    )
- *setDefaultAssertionStatus*  
public synchronized void **setDefaultAssertionStatus**( boolean    )
- *setPackageAssertionStatus*  
public synchronized void **setPackageAssertionStatus**( java.lang.String    , boolean    )
- *setSigners*  
protected final void **setSigners**( java.lang.Class    , java.lang.Object []    )

### 11.2.5 CLASS GlobalContext.PseudoError

---

To test the existence of a class without loading it

#### DECLARATION

---

```
protected class GlobalContext.PseudoError
extends java.lang.Error
```

CONSTRUCTORS

---

- *GlobalContext.PseudoError*  
`protected GlobalContext.PseudoError( )`

METHODS INHERITED FROM CLASS `java.lang.Error`

---

METHODS INHERITED FROM CLASS `java.lang.Throwable`

---

- *fillInStackTrace*  
`public synchronized native Throwable fillInStackTrace( )`
- *getCause*  
`public Throwable getCause( )`
- *getLocalizedMessage*  
`public String getLocalizedMessage( )`
- *getMessage*  
`public String getMessage( )`
- *getOurStackTrace*  
`private synchronized StackTraceElement getOurStackTrace( )`
- *getStackTrace*  
`public StackTraceElement getStackTrace( )`
- *getStackTraceDepth*  
`private native int getStackTraceDepth( )`
- *getStackTraceElement*  
`private native StackTraceElement getStackTraceElement( int )`
- *initCause*  
`public synchronized Throwable initCause( java.lang.Throwable )`
- *printStackTrace*  
`public void printStackTrace( )`
- *printStackTrace*  
`public void printStackTrace( java.io.PrintStream )`
- *printStackTrace*  
`public void printStackTrace( java.io.PrintWriter )`
- *printStackTraceAsCause*  
`private void printStackTraceAsCause( java.io.PrintStream , java.lang.StackTraceElement [] )`
- *printStackTraceAsCause*  
`private void printStackTraceAsCause( java.io.PrintWriter , java.lang.StackTraceElement [] )`
- *setStackTrace*  
`public void setStackTrace( java.lang.StackTraceElement [] )`
- *toString*  
`public String toString( )`
- *writeObject*  
`private synchronized void writeObject( java.io.ObjectOutputStream )`

### 11.2.6 CLASS MethodContext

---

A method method context.

#### DECLARATION

---

```
public class MethodContext
extends koala.dynamicjava.interpreter.context.StaticContext
```

#### CONSTRUCTORS

---

- *MethodContext*  

```
public MethodContext( koala.dynamicjava.interpreter.Interpreter i,
java.lang.Class c, java.lang.Object obj,
koala.dynamicjava.util.ImportationManager im )
```

  - **Usage**
    - \* Creates a new context
  - **Parameters**
    - \* **i** - the interpreter
    - \* **c** - the declaring class of the method
    - \* **obj** - the current object
    - \* **im** - the importation manager

---

- *MethodContext*  

```
public MethodContext( koala.dynamicjava.interpreter.Interpreter i,
java.lang.Class c, java.lang.Object obj, java.util.Set fp )
```

  - **Usage**
    - \* Creates a new context
  - **Parameters**
    - \* **i** - the interpreter
    - \* **c** - the declaring class of the method
    - \* **obj** - the current object
    - \* **fp** - the formal parameters

#### METHODS

---

- *createName*  

```
public Expression createName( koala.dynamicjava.tree.Node node,
koala.dynamicjava.tree.IdentifierToken name )
```

  - **Usage**
    - \* Creates the tree that is associated with the given name
  - **Parameters**
    - \* **node** - the current node
    - \* **name** - the variable name

- **Exceptions**
    - \* `java.lang.IllegalStateException` - if the variable is not defined

---
- *getDefaultQualifier*

```
public Node getDefaultQualifier( koala.dynamicjava.tree.Node node,
                               java.lang.String tname )
```

  - **Usage**
    - \* Returns the default qualifier for this context
  - **Parameters**
    - \* `s` - the qualifier of 'this'

---
- *getHiddenArgument*

```
public Object getHiddenArgument( )
```

  - **Usage**
    - \* Returns the default argument to pass to methods in this context

---
- *getOuterThisName*

```
protected String getOuterThisName( java.lang.Class c )
```

  - **Usage**
    - \* Finds the name of the reference to an outerclass in the given class

---
- *invokeConstructor*

```
public Object invokeConstructor( koala.dynamicjava.tree.ClassAllocation
                                node, java.lang.Object [] args )
```

  - **Usage**
    - \* Invokes a constructor
  - **Parameters**
    - \* `node` - the `ClassAllocation` node
    - \* `args` - the arguments

---
- *invokeConstructor*

```
public Object invokeConstructor( koala.dynamicjava.tree.SimpleAllocation
                                node, java.lang.Object [] args )
```

  - **Usage**
    - \* Invokes a constructor
  - **Parameters**
    - \* `node` - the `SimpleAllocation` node
    - \* `args` - the arguments

---
- *isInnerclass*

```
protected boolean isInnerclass( java.lang.Class ic, java.lang.Class oc )
```

  - **Usage**
    - \* Tests whether an class is an inner class of another
  - **Parameters**
    - \* `ic` - the possibly inner class
    - \* `oc` - the possibly outer class

---

---

- *lookupMethod*

```
public Method lookupMethod( koala.dynamicjava.tree.Node  prefix,
    java.lang.String  mname, java.lang.Class [] params )
```

- **Usage**

- \* Looks for a method

- **Parameters**

- \* **prefix** - the method prefix
    - \* **mname** - the method name
    - \* **params** - the parameter types

- **Exceptions**

- \* `java.lang.NoSuchMethodException` - if the method cannot be found

---

- *setPropertyies*

```
public Class setProperties( koala.dynamicjava.tree.ClassAllocation  node,
    java.lang.Class  c, java.lang.Class [] args, java.util.List  memb )
```

- **Usage**

- \* Sets the properties of a ClassAllocation node

- **Parameters**

- \* **node** - the allocation node
    - \* **c** - the class of the constructor
    - \* **args** - the classes of the arguments of the constructor
    - \* **memb** - the class members

---

- *setPropertyies*

```
public Class setProperties( koala.dynamicjava.tree.SimpleAllocation  node,
    java.lang.Class  c, java.lang.Class [] cargs )
```

- **Usage**

- \* Sets the properties of a SimpleAllocation node

- **Parameters**

- \* **node** - the allocation node
    - \* **c** - the class of the constructor
    - \* **cargs** - the classes of the arguments of the constructor

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.interpreter.context.StaticContext`

---

( in 11.2.9, page 718)

- *classExists*

```
public boolean classExists( java.lang.String  name )
```

- **Usage**

- \* Whether a simple identifier is a class

- **Parameters**

- \* **name** - the identifier

---

- *createName*

```
public Expression createName( koala.dynamicjava.tree.Node  node,
    koala.dynamicjava.tree.IdentifierToken  name )
```



- **Usage**
  - \* Creates the tree that is associated with the given name
- **Parameters**
  - \* **node** - the current node
  - \* **name** - the variable name
- **Exceptions**
  - \* `java.lang.IllegalStateException` - if the variable is not defined

---

- *defineClass*

```
public void defineClass( koala.dynamicjava.tree.TypeDeclaration  node )
```

  - **Usage**
    - \* Defines a class from its syntax tree
  - **Parameters**
    - \* **node** - the class declaration

---

- *defineFunction*

```
public void defineFunction( koala.dynamicjava.tree.MethodDeclaration  node )
```

  - **Usage**
    - \* Defines a MethodDeclaration as a function
  - **Parameters**
    - \* **node** - the function declaration

---

- *fieldExists*

```
protected boolean fieldExists( java.lang.String  name )
```

  - **Usage**
    - \* Whether the given name represents a field in this context
  - **Parameters**
    - \* **name** - the field name

---

- *getDefaultQualifier*

```
public Node getDefaultQualifier( koala.dynamicjava.tree.Node  node )
```

  - **Usage**
    - \* Returns the default qualifier for this context
  - **Parameters**
    - \* **node** - the current node

---

- *getField*

```
public Field getField( java.lang.Class  fc, java.lang.String  fn )
```

  - **Usage**
    - \* Looks for a field
  - **Parameters**
    - \* **fc** - the field class
    - \* **fn** - the field name
  - **Exceptions**
    - \* `java.lang.NoSuchFieldException` - if the field cannot be found
    - \* `koala.dynamicjava.util.AmbiguousFieldException` - if the field is ambiguous

---

- *getModifier*

```
public LeftHandSideModifier getModifier( koala.dynamicjava.tree.SuperFieldAccess  node )
```

  - **Usage**
    - \* Returns the modifier that match the given node

- **Parameters**
    - \* **node** - a tree node

---
- *getSuperField*

```
public Field getSuperField( koala.dynamicjava.tree.Node node, java.lang.String fn )
```

  - **Usage**
    - \* Looks for a field in the super class
  - **Parameters**
    - \* **node** - the current node
    - \* **fn** - the field name
  - **Exceptions**
    - \* `java.lang.NoSuchFieldException` - if the field cannot be found
    - \* `koala.dynamicjava.util.AmbiguousFieldException` - if the field is ambiguous

---
- *isDefined*

```
public boolean isDefined( java.lang.String name )
```

  - **Usage**
    - \* Tests whether a variable is defined in this context
  - **Parameters**
    - \* **name** - the name of the entry
  - **Returns** - false if the variable is undefined

---
- *isInnerClass*

```
protected boolean isInnerClass( java.lang.Class c1, java.lang.Class c2 )
```

  - **Usage**
    - \* Is c1 an inner class of c2?

---
- *lookupClass*

```
public Class lookupClass( java.lang.String cname )
```

  - **Usage**
    - \* Looks for a class
  - **Parameters**
    - \* **cname** - the class name
  - **Exceptions**
    - \* `java.lang.ClassNotFoundException` - if the class cannot be found

---
- *lookupMethod*

```
public Method lookupMethod( koala.dynamicjava.tree.Node prefix, java.lang.String mname, java.lang.Class [] params )
```

  - **Usage**
    - \* Looks for a method
  - **Parameters**
    - \* **prefix** - the method prefix
    - \* **mname** - the method name
    - \* **params** - the parameter types
  - **Exceptions**
    - \* `java.lang.NoSuchMethodException` - if the method cannot be found

---
- *lookupSuperMethod*

```
public Method lookupSuperMethod( koala.dynamicjava.tree.Node node, java.lang.String mname, java.lang.Class [] params )
```

  - **Usage**

- \* Looks for a super method
  - **Parameters**
    - \* **node** - the current node
    - \* **mname** - the method name
    - \* **params** - the parameter types
  - **Exceptions**
    - \* **java.lang.NoSuchMethodException** - if the method cannot be found
- 
- *setAccessFlag*  
 protected void **setAccessFlag**( java.lang.reflect.Member **m** )
    - **Usage**
      - \* Sets the access flag of a member
- 
- *setProperties*  
 public Class **setProperties**( koala.dynamicjava.tree.ClassAllocation **node**,  
 java.lang.Class **c**, java.lang.Class [] **args**, java.util.List **memb** )
    - **Usage**
      - \* Sets the properties of a ClassAllocation node
    - **Parameters**
      - \* **node** - the allocation node
      - \* **c** - the class of the constructor
      - \* **args** - the classes of the arguments of the constructor
      - \* **memb** - the class members

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.interpreter.context.GlobalContext

---

( in 11.2.1, page 658)

- *classExists*  
 public boolean **classExists**( java.lang.String **name** )
    - **Usage**
      - \* Whether a simple identifier is a class
    - **Parameters**
      - \* **name** - the identifier
- 
- *createClassArrayInitializer*  
 protected ArrayInitializer **createClassArrayInitializer**( )
    - **Usage**
      - \* Creates an initializer for the variable class array used to implement inner classes
- 
- *createName*  
 public Expression **createName**( koala.dynamicjava.tree.Node **node**,  
 koala.dynamicjava.tree.IdentifierToken **name** )
    - **Usage**
      - \* Creates the tree that is associated with the given name
    - **Parameters**
      - \* **node** - the current node
      - \* **name** - the variable name
    - **Exceptions**
      - \* **java.lang.IllegalStateException** - if the variable is not defined
- 
- *declareClassImport*  
 public void **declareClassImport**( java.lang.String **cname** )

- **Usage**
  - \* Declares a new single-type-import clause
- **Parameters**
  - \* **cname** - the fully qualified class name
- **Exceptions**
  - \* `java.lang.ClassNotFoundException` - if the class cannot be found

---

- *declarePackageImport*  
`public void declarePackageImport( java.lang.String pkg )`
  - **Usage**
    - \* Declares a new import-on-demand clause
  - **Parameters**
    - \* **pkg** - the package name

---

- *defineClass*  
`public void defineClass( koala.dynamicjava.tree.TypeDeclaration node )`
  - **Usage**
    - \* Defines a class from its syntax tree
  - **Parameters**
    - \* **node** - the class declaration

---

- *defineFunction*  
`public void defineFunction( koala.dynamicjava.tree.MethodDeclaration node )`
  - **Usage**
    - \* Defines a MethodDeclaration as a function
  - **Parameters**
    - \* **node** - the function declaration

---

- *exists*  
`public boolean exists( java.lang.String name )`
  - **Usage**
    - \* Whether a simple identifier represents an existing variable or field or type in this context.
  - **Parameters**
    - \* **name** - the identifier

---

- *getAccessible*  
`public boolean getAccessible( )`
  - **Usage**
    - \* Returns the accessibility state of this context.

---

- *getAdditionalClassLoader*  
`protected ClassLoader getAdditionalClassLoader( )`
  - **Usage**
    - \* Gets the additional class loader

---

- *getCurrentPackage*  
`public String getCurrentPackage( )`
  - **Usage**
    - \* Returns the current package

---

- *getDefaultQualifier*  
`public Node getDefaultQualifier( koala.dynamicjava.tree.Node node )`

- **Usage**
    - \* Returns the default qualifier for this context
  - **Parameters**
    - \* **node** - the current node
- 
- *getDefaultQualifier*

```
public Node getDefaultQualifier( koala.dynamicjava.tree.Node node, java.lang.String tname )
```

    - **Usage**
      - \* Returns the default qualifier for this context
    - **Parameters**
      - \* **node** - the current node
      - \* **tname** - the qualifier of 'this'
- 
- *getField*

```
public Field getField( java.lang.Class fc, java.lang.String fn )
```

    - **Usage**
      - \* Looks for a field
    - **Parameters**
      - \* **fc** - the field class
      - \* **fn** - the field name
    - **Exceptions**
      - \* `java.lang.NoSuchFieldException` - if the field cannot be find
      - \* `koala.dynamicjava.util.AmbiguousFieldException` - if the field is ambiguous
- 
- *getFunctions*

```
public List getFunctions( )
```

    - **Usage**
      - \* Returns the defined functions
- 
- *getHiddenArgument*

```
public Object getHiddenArgument( )
```

    - **Usage**
      - \* Returns the default argument to pass to methods in this context
- 
- *getImportationManager*

```
public ImportationManager getImportationManager( )
```

    - **Usage**
      - \* Returns the importation manager
- 
- *getInterpreter*

```
public Interpreter getInterpreter( )
```

    - **Usage**
      - \* Returns the current interpreter
- 
- *getModifier*

```
public LeftHandSideModifier getModifier( koala.dynamicjava.tree.ObjectFieldAccess node )
```

    - **Usage**
      - \* Returns the modifier that match the given node
    - **Parameters**
      - \* **node** - a tree node
-

- *getModifier*  

```
public LeftHandSideModifier getModifier( koala.dynamicjava.tree.QualifiedName node )
```

  - **Usage**
    - \* Returns the modifier that match the given node
  - **Parameters**
    - \* **node** - a tree node

---
- *getModifier*  

```
public LeftHandSideModifier getModifier( koala.dynamicjava.tree.StaticFieldAccess node )
```

  - **Usage**
    - \* Returns the modifier that match the given node
  - **Parameters**
    - \* **node** - a tree node

---
- *getModifier*  

```
public LeftHandSideModifier getModifier( koala.dynamicjava.tree.SuperFieldAccess node )
```

  - **Usage**
    - \* Returns the modifier that match the given node
  - **Parameters**
    - \* **node** - a tree node

---
- *getPackageName*  

```
protected String getPackageName( java.lang.Class c )
```

  - **Usage**
    - \* Gets the package name for the given class

---
- *getSuperField*  

```
public Field getSuperField( koala.dynamicjava.tree.Node node, java.lang.String fn )
```

  - **Usage**
    - \* Looks for a field in the super class
  - **Parameters**
    - \* **node** - the current node
    - \* **fn** - the field name
  - **Exceptions**
    - \* `java.lang.NoSuchFieldException` - if the field cannot be find
    - \* `koala.dynamicjava.util.AmbiguousFieldException` - if the field is ambiguous

---
- *invokeConstructor*  

```
public Object invokeConstructor( koala.dynamicjava.tree.ClassAllocation node, java.lang.Object [] args )
```

  - **Usage**
    - \* Invokes a constructor
  - **Parameters**
    - \* **node** - the ClassAllocation node
    - \* **args** - the arguments

---
- *invokeConstructor*  

```
public Object invokeConstructor( koala.dynamicjava.tree.SimpleAllocation node, java.lang.Object [] args )
```

- **Usage**
    - \* Invokes a constructor
  - **Parameters**
    - \* **node** - the SimpleAllocation node
    - \* **args** - the arguments
- 
- *isDefined*

```
public boolean isDefined( java.lang.String  name )
```

    - **Usage**
      - \* Tests whether a variable is defined in this context
    - **Parameters**
      - \* **name** - the name of the entry
    - **Returns** - false if the variable is undefined
- 
- *lookupClass*

```
public Class lookupClass( java.lang.String  cname )
```

    - **Usage**
      - \* Looks for a class
    - **Parameters**
      - \* **cname** - the class name
    - **Exceptions**
      - \* `java.lang.ClassNotFoundException` - if the class cannot be found
- 
- *lookupClass*

```
public Class lookupClass( java.lang.String  cname, java.lang.String  ccname )
```

    - **Usage**
      - \* Looks for a class (context-free lookup)
    - **Parameters**
      - \* **cname** - the class name
      - \* **ccname** - the fully qualified name of the context class
    - **Exceptions**
      - \* `java.lang.ClassNotFoundException` - if the class cannot be found
- 
- *lookupConstructor*

```
public Constructor lookupConstructor( java.lang.Class  c, java.lang.Class []
params )
```

    - **Usage**
      - \* Looks for a constructor
    - **Parameters**
      - \* **c** - the class of the constructor
      - \* **params** - the parameter types
    - **Exceptions**
      - \* `java.lang.NoSuchMethodException` - if the constructor cannot be found
- 
- *lookupFunction*

```
public MethodDeclaration lookupFunction( java.lang.String  mname, java.lang.Class
[] params )
```

    - **Usage**
      - \* Looks for a function
    - **Parameters**
      - \* **mname** - the function name
      - \* **params** - the parameter types
    - **Exceptions**

\* koala.dynamicjava.interpreter.context.NoSuchFunctionException - if the function cannot be found

---

- *lookupMethod*

```
public Method lookupMethod( koala.dynamicjava.tree.Node prefix, java.lang.String
mname, java.lang.Class [] params )
```

- **Usage**

- \* Looks for a method

- **Parameters**

- \* **prefix** - the method prefix
    - \* **mname** - the method name
    - \* **params** - the parameter types

- **Exceptions**

- \* java.lang.NoSuchMethodException - if the method cannot be found

---

- *lookupSuperMethod*

```
public Method lookupSuperMethod( koala.dynamicjava.tree.Node node,
java.lang.String mname, java.lang.Class [] params )
```

- **Usage**

- \* Looks for a super method

- **Parameters**

- \* **node** - the current node
    - \* **mname** - the method name
    - \* **params** - the parameter types

- **Exceptions**

- \* java.lang.NoSuchMethodException - if the method cannot be find

---

- *setAccessFlag*

```
protected void setAccessFlag( java.lang.reflect.Member m )
```

- **Usage**

- \* Sets the access flag of a member

---

- *setAccessible*

```
public void setAccessible( boolean accessible )
```

- **Usage**

- \* Allows the scripts to access private fields.

---

- *setAdditionalClassLoaderContainer*

```
public void setAdditionalClassLoaderContainer(
koala.dynamicjava.interpreter.ClassLoaderContainer clc )
```

- **Usage**

- \* Sets the additional class loader container

---

- *setCurrentPackage*

```
public void setCurrentPackage( java.lang.String pkg )
```

- **Usage**

- \* Sets the current package

- **Parameters**

- \* **pkg** - the package name

---

- *setFunctions*

```
public void setFunctions( java.util.List l )
```

- **Usage**

- \* Sets the defined functions



- 
- *setImportationManager*  
 public void **setImportationManager**( koala.dynamicjava.util.ImportationManager im )  
 – **Usage**  
   \* Sets the importation manager

---

  - *setPropertyies*  
 public Class **setPropertyies**( koala.dynamicjava.tree.ClassAllocation node,  
 java.lang.Class c, java.lang.Class [] args, java.util.List memb )  
 – **Usage**  
   \* Sets the properties of a ClassAllocation node  
 – **Parameters**  
   \* **node** - the allocation node  
   \* **c** - the class of the constructor  
   \* **args** - the classes of the arguments of the constructor  
   \* **memb** - the class members

---

  - *setPropertyies*  
 public Class **setPropertyies**( koala.dynamicjava.tree.SimpleAllocation node,  
 java.lang.Class c, java.lang.Class [] cargs )  
 – **Usage**  
   \* Sets the properties of a SimpleAllocation node  
 – **Parameters**  
   \* **node** - the allocation node  
   \* **c** - the class of the constructor  
   \* **cargs** - the classes of the arguments of the constructor

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.context.VariableContext

( in 11.2.10, page 730)

- 
- *define*  
 public void **define**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Defines a new variable in the current scope  
 – **Parameters**  
   \* **name** - the name of the new entry  
   \* **value** - the value of the entry  
 – **Exceptions**  
   \* java.lang.IllegalStateException - if the variable is already defined

---

  - *defineConstant*  
 public void **defineConstant**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Defines a new constant variable in the current scope  
 – **Parameters**  
   \* **name** - the name of the new entry  
   \* **value** - the value of the entry  
 – **Exceptions**  
   \* java.lang.IllegalStateException - if the variable is already defined

---

  - *defineVariables*  
 public void **defineVariables**( java.util.Set vars )

- **Usage**
    - \* Defines the given variables

---
- *enterScope*  
 public void **enterScope**( )
    - **Usage**
      - \* Enters a scope

---
- *enterScope*  
 public void **enterScope**( java.util.Set **entries** )
    - **Usage**
      - \* Enters a scope and defines the given entries to null.
    - **Parameters**
      - \* **entries** - a set of string

---
- *get*  
 public Object **get**( java.lang.String **name** )
    - **Usage**
      - \* Returns the value of a variable with the given name
    - **Parameters**
      - \* **name** - the name of the value to get
    - **Exceptions**
      - \* java.lang.IllegalStateException - if the variable is not defined

---
- *getConstants*  
 public Map **getConstants**( )
    - **Usage**
      - \* Creates a map that contains the constants in this context

---
- *getCurrentScopeVariableNames*  
 public Set **getCurrentScopeVariableNames**( )
    - **Usage**
      - \* Returns the current scope variables (strings) in a set

---
- *getCurrentScopeVariables*  
 public Set **getCurrentScopeVariables**( )
    - **Usage**
      - \* Returns the current scope variables in a set

---
- *isDefinedVariable*  
 public boolean **isDefinedVariable**( java.lang.String **name** )
    - **Usage**
      - \* Tests whether an entry is defined in this context
    - **Parameters**
      - \* **name** - the name of the entry

---
- *isFinal*  
 public boolean **isFinal**( java.lang.String **name** )
    - **Usage**
      - \* Tests whether a variable is final in this context
    - **Parameters**
      - \* **name** - the name of the entry

- **Returns** - false if the variable is not final
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if the variable is not defined
- 
- *leaveScope*

```
public Set leaveScope( )
```

    - **Usage**
      - \* Leaves the current scope
    - **Returns** - the set of variable defined in this scope
- 
- *set*

```
public void set( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a defined variable
    - **Parameters**
      - \* **name** - the name of the entry
      - \* **value** - the value of the entry
    - **Exceptions**
      - \* `java.lang.IllegalStateException` - if the variable is not defined or is final
- 
- *setConstant*

```
public void setConstant( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a constant variable in the current scope
    - **Parameters**
      - \* **name** - the name of the entry
      - \* **value** - the value of the entry
- 
- *setVariable*

```
public void setVariable( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a variable in the current scope
    - **Parameters**
      - \* **name** - the name of the entry
      - \* **value** - the value of the entry

### 11.2.7 CLASS MethodModificationError

---

This error is thrown by a context when it modify the syntax tree

#### DECLARATION

---

```
public class MethodModificationError
extends java.lang.Error
```

#### SERIALIZABLE FIELDS

---

- private Expression expression
  - The modified expression
- private Method method
  - The method

## FIELDS

---

- private Expression expression
  - The modified expression
- private Method method
  - The method

## CONSTRUCTORS

---

- *MethodModificationError*  
**public MethodModificationError( koala.dynamicjava.tree.Expression e,  
java.lang.reflect.Method m )**
  - **Usage**
    - \* Constructs an *MethodModificationError*
  - **Parameters**
    - \* *e* - the new expression
    - \* *m* - the method found

## METHODS

---

- *getExpression*  
**public Expression getExpression( )**
  - **Usage**
    - \* Returns the expression
- *getMethod*  
**public Method getMethod( )**
  - **Usage**
    - \* Returns the method

## METHODS INHERITED FROM CLASS java.lang.Error

---

## METHODS INHERITED FROM CLASS java.lang.Throwable

---

- *fillInStackTrace*  
**public synchronized native Throwable fillInStackTrace( )**
- *getCause*  
**public Throwable getCause( )**
- *getLocalizedMessage*  
**public String getLocalizedMessage( )**

- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

## 11.2.8 CLASS NoSuchFunctionException

---

Thrown when a particular method cannot be found.

### DECLARATION

---

```
public class NoSuchFunctionException
extends java.lang.Exception
```

### CONSTRUCTORS

---

- *NoSuchFunctionException*  
public NoSuchFunctionException( )
    - **Usage**
      - \* Constructs a NoSuchFunctionException without a detail message.
-

- *NoSuchFunctionException*

**public NoSuchFunctionException( java.lang.String s )**

– **Usage**

\* Constructs a NoSuchMethodException with a detail message.

– **Parameters**

\* **s** - the detail message.

#### METHODS INHERITED FROM CLASS java.lang.Exception

---

#### METHODS INHERITED FROM CLASS java.lang.Throwable

---

- *fillInStackTrace*  
public synchronized native Throwable fillInStackTrace( )
- *getCause*  
public Throwable getCause( )
- *getLocalizedMessage*  
public String getLocalizedMessage( )
- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream , java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter , java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

### 11.2.9 CLASS StaticContext

---

A static method context.

#### DECLARATION

---

```
public class StaticContext
extends koala.dynamicjava.interpreter.context.GlobalContext
```

#### CONSTRUCTORS

---

- *StaticContext*  

```
public StaticContext( koala.dynamicjava.interpreter.Interpreter i,
java.lang.Class c, koala.dynamicjava.util.ImportationManager im )
```

  - **Usage**
    - \* Creates a new context
  - **Parameters**
    - \* **i** - the interpreter
    - \* **c** - the declaring class of the method
    - \* **im** - the importation manager

---
- *StaticContext*  

```
public StaticContext( koala.dynamicjava.interpreter.Interpreter i,
java.lang.Class c, java.util.Set fp )
```

  - **Usage**
    - \* Creates a new context
  - **Parameters**
    - \* **i** - the interpreter
    - \* **c** - the declaring class of the method
    - \* **fp** - the formal parameters

#### METHODS

---

- *classExists*  

```
public boolean classExists( java.lang.String name )
```

  - **Usage**
    - \* Whether a simple identifier is a class
  - **Parameters**
    - \* **name** - the identifier

---
- *createName*  

```
public Expression createName( koala.dynamicjava.tree.Node node,
koala.dynamicjava.tree.IdentifierToken name )
```

  - **Usage**

- \* Creates the tree that is associated with the given name
  - **Parameters**
    - \* **node** - the current node
    - \* **name** - the variable name
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if the variable is not defined

---
- *defineClass*

```
public void defineClass( koala.dynamicjava.tree.TypeDeclaration node )
```

  - **Usage**
    - \* Defines a class from its syntax tree
  - **Parameters**
    - \* **node** - the class declaration

---
- *defineFunction*

```
public void defineFunction( koala.dynamicjava.tree.MethodDeclaration node )
```

  - **Usage**
    - \* Defines a MethodDeclaration as a function
  - **Parameters**
    - \* **node** - the function declaration

---
- *fieldExists*

```
protected boolean fieldExists( java.lang.String name )
```

  - **Usage**
    - \* Whether the given name represents a field in this context
  - **Parameters**
    - \* **name** - the field name

---
- *getDefaultQualifier*

```
public Node getDefaultQualifier( koala.dynamicjava.tree.Node node )
```

  - **Usage**
    - \* Returns the default qualifier for this context
  - **Parameters**
    - \* **node** - the current node

---
- *getField*

```
public Field getField( java.lang.Class fc, java.lang.String fn )
```

  - **Usage**
    - \* Looks for a field
  - **Parameters**
    - \* **fc** - the field class
    - \* **fn** - the field name
  - **Exceptions**
    - \* `java.lang.NoSuchFieldException` - if the field cannot be found



---

\* koala.dynamicjava.util.AmbiguousFieldException - if the field is ambiguous

---

- *getModifier*

```
public LeftHandSideModifier getModifier(
    koala.dynamicjava.tree.SuperFieldAccess node )
```

- **Usage**

- \* Returns the modifier that match the given node

- **Parameters**

- \* **node** - a tree node

---

- *getSuperField*

```
public Field getSuperField( koala.dynamicjava.tree.Node node,
    java.lang.String fn )
```

- **Usage**

- \* Looks for a field in the super class

- **Parameters**

- \* **node** - the current node
    - \* **fn** - the field name

- **Exceptions**

- \* java.lang.NoSuchFieldException - if the field cannot be found
    - \* koala.dynamicjava.util.AmbiguousFieldException - if the field is ambiguous

---

- *isDefined*

```
public boolean isDefined( java.lang.String name )
```

- **Usage**

- \* Tests whether a variable is defined in this context

- **Parameters**

- \* **name** - the name of the entry

- **Returns** - false if the variable is undefined

---

- *isInnerClass*

```
protected boolean isInnerClass( java.lang.Class c1, java.lang.Class c2 )
```

- **Usage**

- \* Is c1 an inner class of c2?

---

- *lookupClass*

```
public Class lookupClass( java.lang.String cname )
```

- **Usage**

- \* Looks for a class

- **Parameters**

- \* **cname** - the class name

- **Exceptions**

- \* java.lang.ClassNotFoundException - if the class cannot be found

---

- *lookupMethod*

```
public Method lookupMethod( koala.dynamicjava.tree.Node prefix,
    java.lang.String mname, java.lang.Class [] params )
```

- **Usage**

- \* Looks for a method

- **Parameters**

- \* **prefix** - the method prefix
    - \* **mname** - the method name
    - \* **params** - the parameter types

- **Exceptions**

- \* `java.lang.NoSuchMethodException` - if the method cannot be found

---

- *lookupSuperMethod*

```
public Method lookupSuperMethod( koala.dynamicjava.tree.Node node,
    java.lang.String mname, java.lang.Class [] params )
```

- **Usage**

- \* Looks for a super method

- **Parameters**

- \* **node** - the current node
    - \* **mname** - the method name
    - \* **params** - the parameter types

- **Exceptions**

- \* `java.lang.NoSuchMethodException` - if the method cannot be found

---

- *setAccessFlag*

```
protected void setAccessFlag( java.lang.reflect.Member m )
```

- **Usage**

- \* Sets the access flag of a member

---

- *setProperties*

```
public Class setProperties( koala.dynamicjava.tree.ClassAllocation node,
    java.lang.Class c, java.lang.Class [] args, java.util.List memb )
```

- **Usage**

- \* Sets the properties of a ClassAllocation node

- **Parameters**

- \* **node** - the allocation node
    - \* **c** - the class of the constructor
    - \* **args** - the classes of the arguments of the constructor
    - \* **memb** - the class members

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.interpreter.context.GlobalContext`

---

( in 11.2.1, page 658)

- *classExists*

```
public boolean classExists( java.lang.String name )
```

- **Usage**

- \* Whether a simple identifier is a class
  - **Parameters**
    - \* **name** - the identifier

---
- *createClassArrayInitializer*  
**protected ArrayInitializer createClassArrayInitializer( )**
  - **Usage**
    - \* Creates an initializer for the variable class array used to implement inner classes

---
- *createName*  
**public Expression createName( koala.dynamicjava.tree.Node node, koala.dynamicjava.tree.IdentifierToken name )**
  - **Usage**
    - \* Creates the tree that is associated with the given name
  - **Parameters**
    - \* **node** - the current node
    - \* **name** - the variable name
  - **Exceptions**
    - \* **java.lang.IllegalStateException** - if the variable is not defined

---
- *declareClassImport*  
**public void declareClassImport( java.lang.String cname )**
  - **Usage**
    - \* Declares a new single-type-import clause
  - **Parameters**
    - \* **cname** - the fully qualified class name
  - **Exceptions**
    - \* **java.lang.ClassNotFoundException** - if the class cannot be found

---
- *declarePackageImport*  
**public void declarePackageImport( java.lang.String pkg )**
  - **Usage**
    - \* Declares a new import-on-demand clause
  - **Parameters**
    - \* **pkg** - the package name

---
- *defineClass*  
**public void defineClass( koala.dynamicjava.tree.TypeDeclaration node )**
  - **Usage**
    - \* Defines a class from its syntax tree
  - **Parameters**
    - \* **node** - the class declaration

---
- *defineFunction*  
**public void defineFunction( koala.dynamicjava.tree.MethodDeclaration node )**
  - **Usage**
    - \* Defines a MethodDeclaration as a function
  - **Parameters**
    - \* **node** - the function declaration

---
- *exists*  
**public boolean exists( java.lang.String name )**

- **Usage**
    - \* Whether a simple identifier represents an existing variable or field or type in this context.
  - **Parameters**
    - \* **name** - the identifier

---
- *getAccessible*  
**public boolean getAccessible( )**
  - **Usage**
    - \* Returns the accessibility state of this context.

---
- *getAdditionalClassLoader*  
**protected ClassLoader getAdditionalClassLoader( )**
  - **Usage**
    - \* Gets the additional class loader

---
- *getCurrentPackage*  
**public String getCurrentPackage( )**
  - **Usage**
    - \* Returns the current package

---
- *getDefaultQualifier*  
**public Node getDefaultQualifier( koala.dynamicjava.tree.Node node )**
  - **Usage**
    - \* Returns the default qualifier for this context
  - **Parameters**
    - \* **node** - the current node

---
- *getDefaultQualifier*  
**public Node getDefaultQualifier( koala.dynamicjava.tree.Node node, java.lang.String tname )**
  - **Usage**
    - \* Returns the default qualifier for this context
  - **Parameters**
    - \* **node** - the current node
    - \* **tname** - the qualifier of 'this'

---
- *getField*  
**public Field getField( java.lang.Class fc, java.lang.String fn )**
  - **Usage**
    - \* Looks for a field
  - **Parameters**
    - \* **fc** - the field class
    - \* **fn** - the field name
  - **Exceptions**
    - \* **java.lang.NoSuchFieldException** - if the field cannot be find
    - \* **koala.dynamicjava.util.AmbiguousFieldException** - if the field is ambiguous

---
- *getFunctions*  
**public List getFunctions( )**
  - **Usage**
    - \* Returns the defined functions

---

- *getHiddenArgument*  
 public Object **getHiddenArgument**( )  
 – **Usage**  
 \* Returns the default argument to pass to methods in this context  


---
- *getImportationManager*  
 public ImportationManager **getImportationManager**( )  
 – **Usage**  
 \* Returns the importation manager  


---
- *getInterpreter*  
 public Interpreter **getInterpreter**( )  
 – **Usage**  
 \* Returns the current interpreter  


---
- *getModifier*  
 public LeftHandSideModifier **getModifier**( koala.dynamicjava.tree.ObjectFieldAccess node )  
 – **Usage**  
 \* Returns the modifier that match the given node  
 – **Parameters**  
 \* node - a tree node  


---
- *getModifier*  
 public LeftHandSideModifier **getModifier**( koala.dynamicjava.tree.QualifiedName node )  
 – **Usage**  
 \* Returns the modifier that match the given node  
 – **Parameters**  
 \* node - a tree node  


---
- *getModifier*  
 public LeftHandSideModifier **getModifier**( koala.dynamicjava.tree.StaticFieldAccess node )  
 – **Usage**  
 \* Returns the modifier that match the given node  
 – **Parameters**  
 \* node - a tree node  


---
- *getModifier*  
 public LeftHandSideModifier **getModifier**( koala.dynamicjava.tree.SuperFieldAccess node )  
 – **Usage**  
 \* Returns the modifier that match the given node  
 – **Parameters**  
 \* node - a tree node  


---
- *getPackageName*  
 protected String **getPackageName**( java.lang.Class c )  
 – **Usage**  
 \* Gets the package name for the given class  


---

- *getSuperField*

```
public Field getSuperField( koala.dynamicjava.tree.Node node, java.lang.String fn
)
```

- **Usage**

- \* Looks for a field in the super class

- **Parameters**

- \* **node** - the current node
  - \* **fn** - the field name

- **Exceptions**

- \* `java.lang.NoSuchFieldException` - if the field cannot be find
  - \* `koala.dynamicjava.util.AmbiguousFieldException` - if the field is ambiguous

---

- *invokeConstructor*

```
public Object invokeConstructor( koala.dynamicjava.tree.ClassAllocation node,
java.lang.Object [] args )
```

- **Usage**

- \* Invokes a constructor

- **Parameters**

- \* **node** - the ClassAllocation node
  - \* **args** - the arguments

---

- *invokeConstructor*

```
public Object invokeConstructor( koala.dynamicjava.tree.SimpleAllocation node,
java.lang.Object [] args )
```

- **Usage**

- \* Invokes a constructor

- **Parameters**

- \* **node** - the SimpleAllocation node
  - \* **args** - the arguments

---

- *isDefined*

```
public boolean isDefined( java.lang.String name )
```

- **Usage**

- \* Tests whether a variable is defined in this context

- **Parameters**

- \* **name** - the name of the entry

- **Returns** - false if the variable is undefined

---

- *lookupClass*

```
public Class lookupClass( java.lang.String cname )
```

- **Usage**

- \* Looks for a class

- **Parameters**

- \* **cname** - the class name

- **Exceptions**

- \* `java.lang.ClassNotFoundException` - if the class cannot be found

---

- *lookupClass*

```
public Class lookupClass( java.lang.String cname, java.lang.String ccname )
```

- **Usage**

- \* Looks for a class (context-free lookup)

- **Parameters**

- \* `cname` - the class name
  - \* `ccname` - the fully qualified name of the context class
  - **Exceptions**
    - \* `java.lang.ClassNotFoundException` - if the class cannot be found

---
- *lookupConstructor*

```
public Constructor lookupConstructor( java.lang.Class c, java.lang.Class []
params )
```

  - **Usage**
    - \* Looks for a constructor
  - **Parameters**
    - \* `c` - the class of the constructor
    - \* `params` - the parameter types
  - **Exceptions**
    - \* `java.lang.NoSuchMethodException` - if the constructor cannot be found

---
- *lookupFunction*

```
public MethodDeclaration lookupFunction( java.lang.String mname, java.lang.Class
[] params )
```

  - **Usage**
    - \* Looks for a function
  - **Parameters**
    - \* `mname` - the function name
    - \* `params` - the parameter types
  - **Exceptions**
    - \* `koala.dynamicjava.interpreter.context.NoSuchFunctionException` - if the function cannot be found

---
- *lookupMethod*

```
public Method lookupMethod( koala.dynamicjava.tree.Node prefix, java.lang.String
mname, java.lang.Class [] params )
```

  - **Usage**
    - \* Looks for a method
  - **Parameters**
    - \* `prefix` - the method prefix
    - \* `mname` - the method name
    - \* `params` - the parameter types
  - **Exceptions**
    - \* `java.lang.NoSuchMethodException` - if the method cannot be found

---
- *lookupSuperMethod*

```
public Method lookupSuperMethod( koala.dynamicjava.tree.Node node,
java.lang.String mname, java.lang.Class [] params )
```

  - **Usage**
    - \* Looks for a super method
  - **Parameters**
    - \* `node` - the current node
    - \* `mname` - the method name
    - \* `params` - the parameter types
  - **Exceptions**
    - \* `java.lang.NoSuchMethodException` - if the method cannot be find

---
- *setAccessFlag*

```
protected void setAccessFlag( java.lang.reflect.Member m )
```

- **Usage**
    - \* Sets the access flag of a member

---

- *setAccessible*

```
public void setAccessible( boolean accessible )
```

  - **Usage**
    - \* Allows the scripts to access private fields.

---

- *setAdditionalClassLoaderContainer*

```
public void setAdditionalClassLoaderContainer(
koala.dynamicjava.interpreter.ClassLoaderContainer clc )
```

  - **Usage**
    - \* Sets the additional class loader container

---

- *setCurrentPackage*

```
public void setCurrentPackage( java.lang.String pkg )
```

  - **Usage**
    - \* Sets the current package
  - **Parameters**
    - \* **pkg** - the package name

---

- *setFunctions*

```
public void setFunctions( java.util.List l )
```

  - **Usage**
    - \* Sets the defined functions

---

- *setImportationManager*

```
public void setImportationManager( koala.dynamicjava.util.ImportationManager im )
```

  - **Usage**
    - \* Sets the importation manager

---

- *setProperty*

```
public Class setProperties( koala.dynamicjava.tree.ClassAllocation node,
java.lang.Class c, java.lang.Class [] args, java.util.List memb )
```

  - **Usage**
    - \* Sets the properties of a ClassAllocation node
  - **Parameters**
    - \* **node** - the allocation node
    - \* **c** - the class of the constructor
    - \* **args** - the classes of the arguments of the constructor
    - \* **memb** - the class members

---

- *setProperty*

```
public Class setProperties( koala.dynamicjava.tree.SimpleAllocation node,
java.lang.Class c, java.lang.Class [] cargs )
```

  - **Usage**
    - \* Sets the properties of a SimpleAllocation node
  - **Parameters**
    - \* **node** - the allocation node
    - \* **c** - the class of the constructor
    - \* **cargs** - the classes of the arguments of the constructor



## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.context.VariableContext

( in 11.2.10, page 730)

- *define*  
 public void **define**( java.lang.String **name**, java.lang.Object **value** )
  - **Usage**
    - \* Defines a new variable in the current scope
  - **Parameters**
    - \* **name** - the name of the new entry
    - \* **value** - the value of the entry
  - **Exceptions**
    - \* java.lang.IllegalStateException - if the variable is already defined

---

- *defineConstant*  
 public void **defineConstant**( java.lang.String **name**, java.lang.Object **value** )
  - **Usage**
    - \* Defines a new constant variable in the current scope
  - **Parameters**
    - \* **name** - the name of the new entry
    - \* **value** - the value of the entry
  - **Exceptions**
    - \* java.lang.IllegalStateException - if the variable is already defined

---

- *defineVariables*  
 public void **defineVariables**( java.util.Set **vars** )
  - **Usage**
    - \* Defines the given variables

---

- *enterScope*  
 public void **enterScope**( )
  - **Usage**
    - \* Enters a scope

---

- *enterScope*  
 public void **enterScope**( java.util.Set **entries** )
  - **Usage**
    - \* Enters a scope and defines the given entries to null.
  - **Parameters**
    - \* **entries** - a set of string

---

- *get*  
 public Object **get**( java.lang.String **name** )
  - **Usage**
    - \* Returns the value of a variable with the given name
  - **Parameters**
    - \* **name** - the name of the value to get
  - **Exceptions**
    - \* java.lang.IllegalStateException - if the variable is not defined

---

- *getConstants*  
 public Map **getConstants**( )

- **Usage**
    - \* Creates a map that contains the constants in this context

---

- *getCurrentScopeVariableNames*

```
public Set getCurrentScopeVariableNames( )
```

  - **Usage**
    - \* Returns the current scope variables (strings) in a set

---

- *getCurrentScopeVariables*

```
public Set getCurrentScopeVariables( )
```

  - **Usage**
    - \* Returns the current scope variables in a set

---

- *isDefinedVariable*

```
public boolean isDefinedVariable( java.lang.String name )
```

  - **Usage**
    - \* Tests whether an entry is defined in this context
  - **Parameters**
    - \* **name** - the name of the entry

---

- *isFinal*

```
public boolean isFinal( java.lang.String name )
```

  - **Usage**
    - \* Tests whether a variable is final in this context
  - **Parameters**
    - \* **name** - the name of the entry
  - **Returns** - false if the variable is not final
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if the variable is not defined

---

- *leaveScope*

```
public Set leaveScope( )
```

  - **Usage**
    - \* Leaves the current scope
  - **Returns** - the set of variable defined in this scope

---

- *set*

```
public void set( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a defined variable
  - **Parameters**
    - \* **name** - the name of the entry
    - \* **value** - the value of the entry
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if the variable is not defined or is final

---

- *setConstant*

```
public void setConstant( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a constant variable in the current scope
  - **Parameters**
    - \* **name** - the name of the entry
    - \* **value** - the value of the entry

- 
- *setVariable*  
`public void setVariable( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a variable in the current scope
    - **Parameters**
      - \* **name** - the name of the entry
      - \* **value** - the value of the entry

### 11.2.10 CLASS VariableContext

---

This class encapsulates the behaviour of Java scopes.

#### DECLARATION

---

```
public class VariableContext
extends java.lang.Object
implements SimpleContext
```

#### CONSTRUCTORS

---

- *VariableContext*  
`public VariableContext( )`
    - **Usage**
      - \* Creates a new context initialized with an empty initial scope
- 
- *VariableContext*  
`public VariableContext( java.util.Set entries )`
    - **Usage**
      - \* Creates a new context initialized with the given entries defined in the initial scope.
    - **Parameters**
      - \* **entries** - a set of string

#### METHODS

---

- *define*  
`public void define( java.lang.String name, java.lang.Object value )`
  - **Usage**
    - \* Defines a new variable in the current scope
  - **Parameters**
    - \* **name** - the name of the new entry
    - \* **value** - the value of the entry
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if the variable is already defined

---

- *defineConstant*

```
public void defineConstant( java.lang.String name, java.lang.Object value
)
```

- **Usage**

- \* Defines a new constant variable in the current scope

- **Parameters**

- \* **name** - the name of the new entry
    - \* **value** - the value of the entry

- **Exceptions**

- \* `java.lang.IllegalStateException` - if the variable is already defined

---

- *defineVariables*

```
public void defineVariables( java.util.Set vars )
```

- **Usage**

- \* Defines the given variables

---

- *enterScope*

```
public void enterScope( )
```

- **Usage**

- \* Enters a scope

---

- *enterScope*

```
public void enterScope( java.util.Set entries )
```

- **Usage**

- \* Enters a scope and defines the given entries to null.

- **Parameters**

- \* **entries** - a set of string

---

- *get*

```
public Object get( java.lang.String name )
```

- **Usage**

- \* Returns the value of a variable with the given name

- **Parameters**

- \* **name** - the name of the value to get

- **Exceptions**

- \* `java.lang.IllegalStateException` - if the variable is not defined

---

- *getConstants*

```
public Map getConstants( )
```

- **Usage**

- \* Creates a map that contains the constants in this context

---

- *getCurrentScopeVariableNames*

```
public Set getCurrentScopeVariableNames( )
```

- **Usage**
    - \* Returns the current scope variables (strings) in a set

---
- *getCurrentScopeVariables*

```
public Set getCurrentScopeVariables( )
```

    - **Usage**
      - \* Returns the current scope variables in a set

---
  - *isDefinedVariable*

```
public boolean isDefinedVariable( java.lang.String name )
```

    - **Usage**
      - \* Tests whether an entry is defined in this context
    - **Parameters**
      - \* **name** - the name of the entry

---
  - *isFinal*

```
public boolean isFinal( java.lang.String name )
```

    - **Usage**
      - \* Tests whether a variable is final in this context
    - **Parameters**
      - \* **name** - the name of the entry
    - **Returns** - false if the variable is not final
    - **Exceptions**
      - \* `java.lang.IllegalStateException` - if the variable is not defined

---
  - *leaveScope*

```
public Set leaveScope( )
```

    - **Usage**
      - \* Leaves the current scope
    - **Returns** - the set of variable defined in this scope

---
  - *set*

```
public void set( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a defined variable
    - **Parameters**
      - \* **name** - the name of the entry
      - \* **value** - the value of the entry
    - **Exceptions**
      - \* `java.lang.IllegalStateException` - if the variable is not defined or is final

---
  - *setConstant*

```
public void setConstant( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a constant variable in the current scope

---

– **Parameters**

- \* **name** - the name of the entry
  - \* **value** - the value of the entry
- 

- *setVariable*

```
public void setVariable( java.lang.String  name, java.lang.Object  value )
```

– **Usage**

- \* Sets the value of a variable in the current scope

– **Parameters**

- \* **name** - the name of the entry
- \* **value** - the value of the entry

### 11.2.11 CLASS VariableContext.AbstractVariable

---

To store the variables

#### DECLARATION

---

```
protected abstract static class VariableContext.AbstractVariable
extends java.lang.Object
```

#### FIELDS

---

- public String name
  - The constant name

#### CONSTRUCTORS

---

- *VariableContext.AbstractVariable*  
 protected **VariableContext.AbstractVariable**( )

#### METHODS

---

- *get*  
 public abstract Object get(  
 koala.dynamicjava.interpreter.context.VariableContext ctx )  
  - **Usage**
    - \* Sets the variable in the current scope
- *hashCode*  
 public int hashCode( )  
  - **Usage**

\* Returns the hashCode

---

- *set*

```
public abstract void set(
    koala.dynamicjava.interpreter.context.VariableContext ctx, java.lang.Object
    value )
```

- **Usage**

\* Sets the variable in the current scope

### 11.2.12 CLASS VariableContext.Constant

---

To store the constants

#### DECLARATION

---

```
protected class VariableContext.Constant
extends koala.dynamicjava.interpreter.context.VariableContext.AbstractVariable
```

#### CONSTRUCTORS

---

- *VariableContext.Constant*

```
public VariableContext.Constant( java.lang.String s )
```

- **Usage**

\* Creates a new variable

#### METHODS

---

- *get*

```
public Object get( koala.dynamicjava.interpreter.context.VariableContext ctx
    )
```

- **Usage**

\* Sets the variable in the current scope

---

- *set*

```
public void set( koala.dynamicjava.interpreter.context.VariableContext ctx,
    java.lang.Object value )
```

- **Usage**

\* Sets the variable in the current scope

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.context.VariableContext.AbstractVariable

( in 11.2.11, page 733)

- *get*  
 public abstract Object get( koala.dynamicjava.interpreter.context.VariableContext  
 ctx )  
 – **Usage**  
 \* Sets the variable in the current scope

---

- *hashCode*  
 public int hashCode( )  
 – **Usage**  
 \* Returns the hashCode

---

- *set*  
 public abstract void set( koala.dynamicjava.interpreter.context.VariableContext  
 ctx, java.lang.Object value )  
 – **Usage**  
 \* Sets the variable in the current scope

**11.2.13 CLASS VariableContext.Link**

To store one scope

## DECLARATION

```
protected static class VariableContext.Link
extends java.lang.Object
```

## FIELDS

- public VariableContext.Scope scope  
 – The current scope
- public VariableContext.Scope cscope  
 – The current scope for constants
- public VariableContext.Link next  
 – The next scope



CONSTRUCTORS

---

- *VariableContext.Link*  

```
public VariableContext.Link(
    koala.dynamicjava.interpreter.context.VariableContext.Link next )
```

  - **Usage**  
 \* Creates a new link

**11.2.14 CLASS VariableContext.LinkFactory**

---

To manage the creation of scopes and links

DECLARATION

---

```
protected static class VariableContext.LinkFactory
    extends java.lang.Object
```

CONSTRUCTORS

---

- *VariableContext.LinkFactory*  

```
protected VariableContext.LinkFactory( )
```

METHODS

---

- *createLink*  

```
public static VariableContext.Link createLink(
    koala.dynamicjava.interpreter.context.VariableContext.Link next )
```

  - **Usage**  
 \* Creates a new link
- *recycle*  

```
public static void recycle(
    koala.dynamicjava.interpreter.context.VariableContext.Link l )
```

  - **Usage**  
 \* Notifies the factory to recycle the given link

**11.2.15 CLASS VariableContext.Scope**

---

A table which maps a string with an object

DECLARATION

---

```
protected static class VariableContext.Scope
extends java.lang.Object
```

CONSTRUCTORS

---

- *VariableContext.Scope*  
**public VariableContext.Scope( )**
  - **Usage**
    - \* Creates a new scope

METHODS

---

- *clear*  
**public void clear( )**
  - **Usage**
    - \* Clears this scope

---
- *get*  
**public Object get( java.lang.String key )**
  - **Usage**
    - \* Gets the value of a variable
  - **Returns** - the value or NO\_SUCH\_KEY

---
- *keySet*  
**public Set keySet( )**
  - **Usage**
    - \* Returns a set that contains the keys

---
- *put*  
**public Object put( java.lang.String key, java.lang.Object value )**
  - **Usage**
    - \* Sets a new value for the given variable
  - **Returns** - the old value or NO\_SUCH\_KEY

---
- *rehash*  
**protected void rehash( )**
  - **Usage**
    - \* Rehash the table

### 11.2.16 CLASS VariableContext.Scope.Entry

---

To manage collisions

#### DECLARATION

---

```
protected static class VariableContext.Scope.Entry
extends java.lang.Object
```

#### FIELDS

---

- public int hash
  - The hash code
- public String key
  - The variable
- public Object value
  - The value
- public VariableContext.Scope.Entry next
  - The next entry

#### CONSTRUCTORS

---

- *VariableContext.Scope.Entry*  
 public **VariableContext.Scope.Entry**( int hash, java.lang.String key,  
 java.lang.Object value,  
 koala.dynamicjava.interpreter.context.VariableContext.Scope.Entry next )  
  - **Usage**
    - \* Creates a new entry

### 11.2.17 CLASS VariableContext.Scope.EntryFactory

---

To create an entry

#### DECLARATION

---

```
protected static class VariableContext.Scope.EntryFactory
extends java.lang.Object
```

CONSTRUCTORS

---

- *VariableContext.Scope.EntryFactory*  
protected **VariableContext.Scope.EntryFactory**( )

METHODS

---

- *createEntry*  
public static **VariableContext.Scope.Entry** createEntry( int hash,  
java.lang.String key, java.lang.Object value,  
koala.dynamicjava.interpreter.context.VariableContext.Scope.Entry next )  
  
– **Usage**  
\* Creates a new entry

**11.2.18 CLASS VariableContext.Variable**

---

To store the variables

DECLARATION

---

```
protected static class VariableContext.Variable
extends koala.dynamicjava.interpreter.context.VariableContext.AbstractVariable
```

CONSTRUCTORS

---

- *VariableContext.Variable*  
public **VariableContext.Variable**( java.lang.String s )  
  
– **Usage**  
\* Creates a new variable

METHODS

---

- *get*  
public **Object** get( koala.dynamicjava.interpreter.context.VariableContext ctx  
)  
  
– **Usage**  
\* Sets the variable in the current scope
- 
- *set*  
public void **set**( koala.dynamicjava.interpreter.context.VariableContext ctx,  
java.lang.Object value )  
  
– **Usage**  
\* Sets the variable in the current scope

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.context.VariableContext.AbstractVariable

---

( in 11.2.11, page 733)

- *get*  
public abstract Object get( koala.dynamicjava.interpreter.context.VariableContext  
ctx )
  - **Usage**
    - \* Sets the variable in the current scope

---
- *hashCode*  
public int hashCode( )
  - **Usage**
    - \* Returns the hashCode

---
- *set*  
public abstract void set( koala.dynamicjava.interpreter.context.VariableContext  
ctx, java.lang.Object value )
  - **Usage**
    - \* Sets the variable in the current scope

## Chapter 12

# Package koala.dynamicjava.classinfo

Package Contents

Page

### Interfaces

<b>ClassFinder</b> .....	743
<i>The instances of the classes that implements this interface are used to find the fully qualified name of classes and to manage the loading of these classes.</i>	
<b>ClassInfo</b> .....	744
<i>The instances of the classes that implement this interface provide informations about classes.</i>	
<b>ConstructorInfo</b> .....	746
<i>The instances of the classes that implement this interface provide informations about constructors.</i>	
<b>FieldInfo</b> .....	747
<i>The instances of the classes that implement this interface provide informations about class fields.</i>	
<b>MethodInfo</b> .....	747
<i>The instances of the classes that implement this interface provide informations about methods.</i>	

### Classes

<b>ClassInfoUtilities</b> .....	748
<i>This class contains a collection of utility methods for reflection.</i>	
<b>JavaClassInfo</b> .....	751
<i>The instances of this class provides informations about class compiled to JVM bytecode.</i>	
<b>JavaConstructorInfo</b> .....	755
<i>The instances of this class provides informations about class constructors compiled to JVM bytecode.</i>	
<b>JavaFieldInfo</b> .....	756
<i>The instances of this class provides informations about class fields compiled to JVM bytecode.</i>	
<b>JavaMethodInfo</b> .....	757
<i>The instances of this class provides informations about class methods compiled to JVM bytecode.</i>	
<b>TreeClassInfo</b> .....	759
<i>The instances of this class provides informations about classes not yet compiled to JVM bytecode and represented by a syntax tree</i>	
<b>TreeClassInfo.MembersVisitor</b> .....	764
<i>To initialize the ClassInfo</i>	
<b>TreeConstructorInfo</b> .....	778

	<i>The instances of this class provides informations about class constructors not yet compiled to JVM bytecode.</i>	
<b>TreeFieldInfo</b> .....		780
	<i>The instances of this class provides informations about class fields not yet compiled to JVM bytecode.</i>	
<b>TreeMethodInfo</b> .....		781
	<i>The instances of this class provides informations about class methods not yet compiled to JVM bytecode.</i>	
<b>TypeVisitor</b> .....		784
	<i>The instances of this class are used to get the ClassInfo that match a type node of a syntax tree</i>	

---

## 12.1 Interfaces

### 12.1.1 INTERFACE ClassFinder

The instances of the classes that implements this interface are used to find the fully qualified name of classes and to manage the loading of these classes.

#### DECLARATION

```
public interface ClassFinder
```

#### METHODS

- *addClassInfo*

```
public ClassInfo addClassInfo( java.lang.String  cname,
                               koala.dynamicjava.tree.TypeDeclaration  decl )
```

- **Usage**

- \* Adds a type declaration in the class info list

- **Parameters**

- \* **cname** - the name of the class
  - \* **decl** - the type declaration

- *getCurrentPackage*

```
public String getCurrentPackage( )
```

- **Usage**

- \* Returns the current package

- *lookupClass*

```
public ClassInfo lookupClass( java.lang.String  cname )
```

- **Usage**

- \* Loads the class info that match the given name in the source file

- **Parameters**

- \* **cname** - the name of the class to find

- **Returns** - the class info

- **Exceptions**

- \* `java.lang.ClassNotFoundException` - if the class cannot be loaded

- *lookupClass*

```
public ClassInfo lookupClass( java.lang.String  cname,
                               koala.dynamicjava.classinfo.ClassInfo  cinfo )
```

- **Usage**

- \* Loads the class info that match the given name in the source file

- **Parameters**

- \* **cname** - the name of the class to find



- \* **cinfo** - the context where 'cname' was found
- **Returns** - the class info
- **Exceptions**
  - \* `java.lang.ClassNotFoundException` - if the class cannot be loaded

### 12.1.2 INTERFACE ClassInfo

---

The instances of the classes that implement this interface provide informations about classes.

#### DECLARATION

---

```
public interface ClassInfo
```

#### METHODS

---

- *getAnonymousDeclaringClass*  

```
public ClassInfo getAnonymousDeclaringClass( )
```

  - **Usage**
    - \* Returns the declaring class of an anonymous class or null

---
- *getArrayType*  

```
public ClassInfo getArrayType( )
```

  - **Usage**
    - \* Returns the array type that contains elements of this class

---
- *getComponentType*  

```
public ClassInfo getComponentType( )
```

  - **Usage**
    - \* Returns the component type of this array type
  - **Exceptions**
    - \* `java.lang.IllegalStateException` - if this type do not represent an array

---
- *getConstructors*  

```
public ConstructorInfo getConstructors( )
```

  - **Usage**
    - \* Returns the constructor infos for the current class

---
- *getDeclaredClasses*  

```
public ClassInfo getDeclaredClasses( )
```

  - **Usage**
    - \* Returns the classes and interfaces declared as members of the class represented by this ClassInfo object.

---

- *getDeclaringClass*  
 public ClassInfo **getDeclaringClass**( )  
 – **Usage**  
 \* Returns the declaring class or null  


---
- *getFields*  
 public FieldInfo **getFields**( )  
 – **Usage**  
 \* Returns the field infos for the current class  


---
- *getInterfaces*  
 public ClassInfo **getInterfaces**( )  
 – **Usage**  
 \* Returns the class infos of the interfaces implemented by the class this info represents  


---
- *getJavaClass*  
 public Class **getJavaClass**( )  
 – **Usage**  
 \* Returns the underlying class  


---
- *getMethods*  
 public MethodInfo **getMethods**( )  
 – **Usage**  
 \* Returns the method infos for the current class  


---
- *getModifiers*  
 public int **getModifiers**( )  
 – **Usage**  
 \* Returns the modifiers flags  


---
- *getName*  
 public String **getName**( )  
 – **Usage**  
 \* Returns the fully qualified name of the underlying class  


---
- *getSuperclass*  
 public ClassInfo **getSuperclass**( )  
 – **Usage**  
 \* Returns the class info of the superclass of the class represented by this info  


---
- *isArray*  
 public boolean **isArray**( )  
 – **Usage**

- \* Whether this object represents an array
- • *isCompilable*  
 public boolean **isCompilable**( )
  - **Usage**
    - \* Whether the underlying class needs compilation

---
- *isInterface*  
 public boolean **isInterface**( )
  - **Usage**
    - \* Whether this object represents an interface

---
- *isPrimitive*  
 public boolean **isPrimitive**( )
  - **Usage**
    - \* Whether this object represents a primitive type

---
- *setCompilable*  
 public void **setCompilable**( boolean b )
  - **Usage**
    - \* Sets the compilable property

### 12.1.3 INTERFACE ConstructorInfo

---

The instances of the classes that implement this interface provide informations about constructors.

#### DECLARATION

---

public interface ConstructorInfo

#### METHODS

---

- *getExceptionTypes*  
 public ClassInfo **getExceptionTypes**( )
  - **Usage**
    - \* Returns an array of class infos that represent the types of the exceptions declared to be thrown by the underlying constructor

---
- *getModifiers*  
 public int **getModifiers**( )
  - **Usage**
    - \* Returns the modifiers for the constructor represented by this object

---

- *getParameterTypes*

**public ClassInfo getParameterTypes( )**

– **Usage**

- \* Returns an array of class infos that represent the parameter types, in declaration order, of the constructor represented by this object

### 12.1.4 INTERFACE **FieldInfo**

---

The instances of the classes that implement this interface provide informations about class fields.

#### DECLARATION

---

```
public interface FieldInfo
```

#### METHODS

---

- *getModifiers*

**public int getModifiers( )**

– **Usage**

- \* Returns the modifiers for the field represented by this object
- 

- *getName*

**public String getName( )**

– **Usage**

- \* Returns the fully qualified name of the underlying field
- 

- *getType*

**public ClassInfo getType( )**

– **Usage**

- \* Returns the type of the underlying field

### 12.1.5 INTERFACE **MethodInfo**

---

The instances of the classes that implement this interface provide informations about methods.

#### DECLARATION

---

```
public interface MethodInfo
```

## METHODS

• *getExceptionTypes*

```
public ClassInfo getExceptionTypes( )
```

## – Usage

\* Returns an array of class infos that represent the types of the exceptions declared to be thrown by the underlying method

• *getModifiers*

```
public int getModifiers( )
```

## – Usage

\* Returns the modifiers for the method represented by this object

• *getName*

```
public String getName( )
```

## – Usage

\* Returns the name of the underlying method

• *getParameterTypes*

```
public ClassInfo getParameterTypes( )
```

## – Usage

\* Returns an array of class infos that represent the parameter types, in declaration order, of the method represented by this object

• *getReturnType*

```
public ClassInfo getReturnType( )
```

## – Usage

\* Returns a Class object that represents the return type of the method represented by this object

## 12.2 Classes

### 12.2.1 CLASS ClassInfoUtilities

This class contains a collection of utility methods for reflection.

## DECLARATION

```
public class ClassInfoUtilities
extends java.lang.Object
```

## CONSTRUCTORS

---

- *ClassInfoUtilities*

**private ClassInfoUtilities( )**

- **Usage**

- \* This class contains only static methods, so it is not useful to create instances of it or to extend it.

## METHODS

---

- *getConstructors*

**private static List getConstructors( koala.dynamicjava.classinfo.ClassInfo cl, int params )**

- **Usage**

- \* Gets all the constructors in the given class or super classes, even the redefined constructors are returned.

- **Parameters**

- \* **cl** - the class where the constructor was declared
- \* **params** - the number of parameters

- **Returns** - a list that contains the found constructors, an empty list if no matching constructor was found.

---

- *getField*

**public static FieldInfo getField( koala.dynamicjava.classinfo.ClassInfo cl, java.lang.String name )**

- **Usage**

- \* Returns a field with the given name declared in the given class or in the superclasses of the given class

- **Parameters**

- \* **cl** - the class where the field must look for the field
  - \* **name** - the name of the field
- 

- *getMethods*

**public static List getMethods( koala.dynamicjava.classinfo.ClassInfo cl, java.lang.String name, int params )**

- **Usage**

- \* Gets all the methods with the given name in the given class or super classes. Even the redefined methods are returned.

- **Parameters**

- \* **cl** - the class where the method was declared
- \* **name** - the name of the method
- \* **params** - the number of parameters

- **Returns** - a list that contains the found methods, an empty list if no matching method was found.

---

- *getOuterField*

```
public static FieldInfo getOuterField( koala.dynamicjava.classinfo.ClassInfo
cl, java.lang.String name )
```

- **Usage**

- \* Returns a field with the given name declared in one of the outer classes of the given class

- **Parameters**

- \* **cl** - the inner class
    - \* **name** - the name of the field

---

- *hasCompatibleSignatures*

```
private static boolean hasCompatibleSignatures(
koala.dynamicjava.classinfo.ClassInfo [] a1,
koala.dynamicjava.classinfo.ClassInfo [] a2 )
```

- **Usage**

- \* For each element of the given arrays, tests if the first array element is assignable from the second array element. The two arrays are assumed to have the same length.

---

- *isAncestorOf*

```
private static boolean isAncestorOf( koala.dynamicjava.classinfo.ClassInfo
cl, koala.dynamicjava.classinfo.ClassInfo c2 )
```

- **Usage**

- \* Is c1 an ancestor of c2 ?

---

- *isAssignableFrom*

```
public static boolean isAssignableFrom(
koala.dynamicjava.classinfo.ClassInfo c1,
koala.dynamicjava.classinfo.ClassInfo c2 )
```

- **Usage**

- \* Tests whether c1 is assignable from c2. This function works on every types (primitive or not)

- **Parameters**

- \* **c1** - a class info
    - \* **c2** - a class info

---

- *isInterfaceOf*

```
private static boolean isInterfaceOf( koala.dynamicjava.classinfo.ClassInfo
cl, koala.dynamicjava.classinfo.ClassInfo c2 )
```

- **Usage**

- \* Is c1 an interface of c2 ?

---

- *lookupConstructor*

```
public static ConstructorInfo lookupConstructor(
koala.dynamicjava.classinfo.ClassInfo cl,
koala.dynamicjava.classinfo.ClassInfo [] ac )
```

---

– **Usage**

- \* Looks for a constructor in the given class or in super classes of this class.

– **Parameters**

- \* **cl** - the class of which the constructor is a member
  - \* **ac** - the arguments classes (possibly not the exact declaring classes)
- 

- *lookupMethod*

```
public static MethodInfo lookupMethod(
koala.dynamicjava.classinfo.ClassInfo cl, java.lang.String name,
koala.dynamicjava.classinfo.ClassInfo [] ac )
```

– **Usage**

- \* Looks for a method in the given class or in super classes of this class.

– **Parameters**

- \* **cl** - the class of which the method is a member
  - \* **name** - the name of the method
  - \* **ac** - the arguments classes (possibly not the exact declaring classes)
- 

- *lookupOuterMethod*

```
public static MethodInfo lookupOuterMethod(
koala.dynamicjava.classinfo.ClassInfo cl, java.lang.String name,
koala.dynamicjava.classinfo.ClassInfo [] ac )
```

– **Usage**

- \* Looks up for a method in an outer classes of this class.

– **Parameters**

- \* **cl** - the inner class
  - \* **name** - the name of the method
  - \* **ac** - the arguments classes (possibly not the exact declaring classes)
- 

- *selectTheMostSpecificConstructor*

```
private static ConstructorInfo selectTheMostSpecificConstructor(
koala.dynamicjava.classinfo.ConstructorInfo m1,
koala.dynamicjava.classinfo.ConstructorInfo m2 )
```

– **Usage**

- \* Returns the constructor with the most specific signature.
- 

- *selectTheMostSpecificMethod*

```
private static MethodInfo selectTheMostSpecificMethod(
koala.dynamicjava.classinfo.MethodInfo m1,
koala.dynamicjava.classinfo.MethodInfo m2 )
```

– **Usage**

- \* Returns the method with the most specific signature.

## 12.2.2 CLASS *JavaClassInfo*

---

The instances of this class provides informations about class compiled to JVM bytecode.



DECLARATION

---

```
public class JavaClassInfo
extends java.lang.Object
implements ClassInfo
```

FIELDS

---

- public static final JavaClassInfo BOOLEAN
  - The boolean info
- public static final JavaClassInfo INT
  - The int info
- public static final JavaClassInfo LONG
  - The long info
- public static final JavaClassInfo FLOAT
  - The float info
- public static final JavaClassInfo DOUBLE
  - The double info
- public static final JavaClassInfo STRING
  - The string info
- public static final JavaClassInfo CLASS
  - The Class info
- private Class javaClass
  - The underlying class

CONSTRUCTORS

---

- *JavaClassInfo*  
 public **JavaClassInfo**( java.lang.Class c )
  - **Usage**
    - \* Creates a new class info
  - **Parameters**
    - \* c - the java class

---
- *JavaClassInfo*  
 public **JavaClassInfo**( koala.dynamicjava.classinfo.JavaClassInfo c )
  - **Usage**
    - \* Creates a new class info representing an array
  - **Parameters**
    - \* c - the java class

METHODS

---

- *equals*  
public boolean equals( java.lang.Object obj )
  - **Usage**
    - \* Indicates whether some other object is "equal to" this one

---
- *getAnonymousDeclaringClass*  
public ClassInfo getAnonymousDeclaringClass( )
  - **Usage**
    - \* Returns the declaring class of an anonymous class or null

---
- *getArrayType*  
public ClassInfo getArrayType( )
  - **Usage**
    - \* Returns the array type that contains elements of this class

---
- *getComponentType*  
public ClassInfo getComponentType( )
  - **Usage**
    - \* Returns the component type of this array type
  - **Exceptions**
    - \* java.lang.IllegalStateException - if this type do not represent an array

---
- *getConstructors*  
public ConstructorInfo getConstructors( )
  - **Usage**
    - \* Returns the constructor infos for the current class

---
- *getDeclaredClasses*  
public ClassInfo getDeclaredClasses( )
  - **Usage**
    - \* Returns the classes and interfaces declared as members of the class represented by this ClassInfo object.

---
- *getDeclaringClass*  
public ClassInfo getDeclaringClass( )
  - **Usage**
    - \* Returns the declaring class or null

---
- *getFields*  
public FieldInfo getFields( )
  - **Usage**
    - \* Returns the field infos for the current class

- 
- *getInterfaces*  
`public ClassInfo getInterfaces( )`
    - **Usage**
      - \* Returns the class infos of the interfaces implemented by the class this info represents
- 
- *getJavaClass*  
`public Class getJavaClass( )`
    - **Usage**
      - \* Returns the underlying class
- 
- *getMethods*  
`public MethodInfo getMethods( )`
    - **Usage**
      - \* Returns the method infos for the current class
- 
- *getModifiers*  
`public int getModifiers( )`
    - **Usage**
      - \* Returns the modifiers flags
- 
- *getName*  
`public String getName( )`
    - **Usage**
      - \* Returns the fully qualified name of the underlying class
- 
- *getSuperclass*  
`public ClassInfo getSuperclass( )`
    - **Usage**
      - \* Returns the class info of the superclass of the class represented by this info
- 
- *isArray*  
`public boolean isArray( )`
    - **Usage**
      - \* Whether this object represents an array
- 
- *isCompilable*  
`public boolean isCompilable( )`
    - **Usage**
      - \* Whether the underlying class needs compilation
- 
- *isInterface*  
`public boolean isInterface( )`

- **Usage**
    - \* Whether this object represents an interface
- 
- *isPrimitive*

```
public boolean isPrimitive( )
```

    - **Usage**
      - \* Whether this object represents a primitive type
- 
- *setCompilable*

```
public void setCompilable( boolean b )
```

    - **Usage**
      - \* Sets the compilable property

### 12.2.3 CLASS *JavaConstructorInfo*

The instances of this class provides informations about class constructors compiled to JVM bytecode.

#### DECLARATION

```
public class JavaConstructorInfo
extends java.lang.Object
implements ConstructorInfo
```

#### FIELDS

- private Constructor javaConstructor
  - The underlying constructor

#### CONSTRUCTORS

- *JavaConstructorInfo*

```
public JavaConstructorInfo( java.lang.reflect.Constructor f )
```

  - **Usage**
    - \* Creates a new class info
  - **Parameters**
    - \* *f* - the java constructor

METHODS

---

- *equals*  
`public boolean equals( java.lang.Object obj )`
  - **Usage**
    - \* Indicates whether some other object is "equal to" this one

---
- *getExceptionTypes*  
`public ClassInfo getExceptionTypes( )`
  - **Usage**
    - \* Returns an array of Class infos that represent the types of the exceptions declared to be thrown by the underlying constructor

---
- *getModifiers*  
`public int getModifiers( )`
  - **Usage**
    - \* Returns the modifiers for the constructor represented by this object

---
- *getParameterTypes*  
`public ClassInfo getParameterTypes( )`
  - **Usage**
    - \* Returns an array of class infos that represent the parameter types, in declaration order, of the constructor represented by this object

#### 12.2.4 CLASS *JavaFieldInfo*

---

The instances of this class provides informations about class fields compiled to JVM bytecode.

DECLARATION

---

```
public class JavaFieldInfo
extends java.lang.Object
implements FieldInfo
```

FIELDS

---

- private Field *javaField*
  - The underlying field

## CONSTRUCTORS

---

- *JavaFieldInfo*  
`public JavaFieldInfo( java.lang.reflect.Field f )`
  - **Usage**
    - \* Creates a new class info
  - **Parameters**
    - \* `f` - the java field

## METHODS

---

- *equals*  
`public boolean equals( java.lang.Object obj )`
  - **Usage**
    - \* Indicates whether some other object is "equal to" this one
- *getModifiers*  
`public int getModifiers( )`
  - **Usage**
    - \* Returns the modifiers for the field represented by this object
- *getName*  
`public String getName( )`
  - **Usage**
    - \* Returns the fully qualified name of the underlying field
- *getType*  
`public ClassInfo getType( )`
  - **Usage**
    - \* Returns the type of the underlying field

### 12.2.5 CLASS **JavaMethodInfo**

---

The instances of this class provides informations about class methods compiled to JVM bytecode.

## DECLARATION

---

```
public class JavaMethodInfo
extends java.lang.Object
implements MethodInfo
```

## FIELDS

---

- private Method javaMethod
  - The underlying method
- private ClassInfo parameters
  - The parameters types
- private ClassInfo exceptions
  - The exception types

## CONSTRUCTORS

---

- *JavaMethodInfo*  
`public JavaMethodInfo( java.lang.reflect.Method f )`
  - **Usage**
    - \* Creates a new method info
  - **Parameters**
    - \* **f** - the java method

## METHODS

---

- *equals*  
`public boolean equals( java.lang.Object obj )`
  - **Usage**
    - \* Indicates whether some other object is "equal to" this one
- *getExceptionTypes*  
`public ClassInfo getExceptionTypes( )`
  - **Usage**
    - \* Returns an array of Class infos that represent the types of the exceptions declared to be thrown by the underlying method
- *getModifiers*  
`public int getModifiers( )`
  - **Usage**
    - \* Returns the modifiers for the method represented by this object
- *getName*  
`public String getName( )`
  - **Usage**
    - \* Returns the name of the underlying method

- *getParameterTypes*

**public ClassInfo getParameterTypes( )**

- **Usage**

- \* Returns an array of class infos that represent the parameter types, in declaration order, of the method represented by this object

- *getReturnType*

**public ClassInfo getReturnType( )**

- **Usage**

- \* Returns a Class object that represents the return type of the method represented by this object

## 12.2.6 CLASS TreeClassInfo

---

The instances of this class provides informations about classes not yet compiled to JVM bytecode and represented by a syntax tree

### DECLARATION

---

```
public class TreeClassInfo
extends java.lang.Object
implements ClassInfo
```

### FIELDS

---

- private static final String DECLARING\_CLASS
  - The declaringClass property is defined for each inner class/interface declaration It contains a TypeDeclaration
- public static final String ANONYMOUS\_DECLARING\_CLASS
  - The declaringClass property is defined for each anonymous inner class/interface declaration It contains a TypeDeclaration
- private static final String TREE\_VISITED
  - This property is used to ensure that the modifications on the tree are not done twice
- private TypeDeclaration classTree
  - The abstract syntax tree of this class
- private ClassFinder classFinder
  - The class finder for this class
- private int dimension
  - The dimension of this type



- private String name
  - The full class name
- private ClassInfo superclass
  - The class info of the superclass of the class represented by this field
- private boolean interfaceInfo
  - Whether this class is an interface
- private ClassInfo interfaces
  - The interfaces
- private Map fields
  - The fields
- private Map methods
  - The methods
- private List constructors
  - The constructors
- private List classes
  - The declared classes
- private boolean compilable
  - The compilable property value
- private int methodCount
  - The method count

## CONSTRUCTORS

---

- *TreeClassInfo*

```
public TreeClassInfo( koala.dynamicjava.classinfo.TreeClassInfo  ci )
```

    - **Usage**
      - \* Creates a new array class info
    - **Parameters**
      - \* ci - the class info
- 
- *TreeClassInfo*

```
public TreeClassInfo( koala.dynamicjava.tree.TypeDeclaration  cd,
koala.dynamicjava.classinfo.ClassFinder  cf )
```

    - **Usage**
      - \* Creates a new class info
    - **Parameters**
      - \* cd - the class declaration
      - \* cf - the class finder

METHODS

---

- *equals*  
`public boolean equals( java.lang.Object obj )`  
  - **Usage**  
\* Indicates whether some other object is "equal to" this one

---
- *fullName*  
`private String fullName( )`  
  - **Usage**  
\* Returns the full name of this class

---
- *getAnonymousDeclaringClass*  
`public ClassInfo getAnonymousDeclaringClass( )`  
  - **Usage**  
\* Returns the declaring class of an anonymous class or null

---
- *getArrayType*  
`public ClassInfo getArrayType( )`  
  - **Usage**  
\* Returns the array type that contains elements of this class

---
- *getClassFinder*  
`public ClassFinder getClassFinder( )`  
  - **Usage**  
\* Returns the class finder

---
- *getComponentType*  
`public ClassInfo getComponentType( )`  
  - **Usage**  
\* Returns the component type of this array type
  - **Exceptions**  
\* `java.lang.IllegalStateException` - if this type do not represent an array

---
- *getConstructors*  
`public ConstructorInfo getConstructors( )`  
  - **Usage**  
\* Returns the constructor infos for the current class

---
- *getDeclaredClasses*  
`public ClassInfo getDeclaredClasses( )`  
  - **Usage**  
\* Returns the classes and interfaces declared as members of the class represented by this `ClassInfo` object.

- 
- *getDeclaringClass*  
 public ClassInfo **getDeclaringClass**( )  
 – **Usage**  
 \* Returns the declaring class or null

---

  - *getField*  
 public FieldInfo **getField**( koala.dynamicjava.tree.FieldDeclaration node )  
 – **Usage**  
 \* Returns the field represented by the given node  
 – **Parameters**  
 \* node - the node that represents the field

---

  - *getFields*  
 public FieldInfo **getFields**( )  
 – **Usage**  
 \* Returns the field infos for the current class

---

  - *getInterfaces*  
 public ClassInfo **getInterfaces**( )  
 – **Usage**  
 \* Returns the class infos of the interfaces implemented by the class this info represents  
 – **Exceptions**  
 \* java.lang.NoClassDefFoundError - if an interface cannot be loaded

---

  - *getJavaClass*  
 public Class **getJavaClass**( )  
 – **Usage**  
 \* Returns the underlying class

---

  - *getMethod*  
 public MethodInfo **getMethod**( koala.dynamicjava.tree.MethodDeclaration node )  
 – **Usage**  
 \* Returns the method represented by the given node  
 – **Parameters**  
 \* node - the node that represents the method

---

  - *getMethods*  
 public MethodInfo **getMethods**( )  
 – **Usage**  
 \* Returns the method infos for the current class

---

- *getModifiers*  
`public int getModifiers( )`
  - **Usage**
    - \* Returns the modifiers flags

---
- *getName*  
`public String getName( )`
  - **Usage**
    - \* Returns the fully qualified name of the underlying class

---
- *getNestingLevel*  
`private int getNestingLevel( )`
  - **Usage**
    - \* Returns the nesting level of the class

---
- *getSuperclass*  
`public ClassInfo getSuperclass( )`
  - **Usage**
    - \* Returns the class info of the superclass of the class represented by this class
  - **Exceptions**
    - \* `java.lang.NoClassDefFoundError` - if the class cannot be loaded

---
- *getTypeDeclaration*  
`public TypeDeclaration getTypeDeclaration( )`
  - **Usage**
    - \* Returns the abstract syntax tree

---
- *isArray*  
`public boolean isArray( )`
  - **Usage**
    - \* Whether this object represents an array

---
- *isCompilable*  
`public boolean isCompilable( )`
  - **Usage**
    - \* Whether the underlying class needs compilation

---
- *isInterface*  
`public boolean isInterface( )`
  - **Usage**
    - \* Whether this object represents an interface

---
- *isPrimitive*  
`public boolean isPrimitive( )`

- **Usage**
    - \* Whether this object represents a primitive type
- 
- *lookupClass*

```
private ClassInfo lookupClass( java.lang.String s )
```

    - **Usage**
      - \* Looks for a class from its name
    - **Parameters**
      - \* **s** - the name of the class to find
    - **Exceptions**
      - \* `java.lang.NoClassDefFoundError` - if the class cannot be loaded
- 
- *lookupClass*

```
private ClassInfo lookupClass( java.lang.String s,
koala.dynamicjava.classinfo.ClassInfo c )
```

    - **Usage**
      - \* Looks for a class from its name
    - **Parameters**
      - \* **s** - the name of the class to find
      - \* **c** - the context
    - **Exceptions**
      - \* `java.lang.NoClassDefFoundError` - if the class cannot be loaded
- 
- *setCompilable*

```
public void setCompilable( boolean b )
```

    - **Usage**
      - \* Sets the compilable property

### 12.2.7 CLASS `TreeClassInfo.MembersVisitor`

---

To initialize the `ClassInfo`

#### DECLARATION

---

```
private class TreeClassInfo.MembersVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

#### CONSTRUCTORS

---

- *TreeClassInfo.MembersVisitor*

```
TreeClassInfo.MembersVisitor( )
```

  - **Usage**
    - \* Creates a new members visitor and iterate over the members of the class represented by this `ClassInfo`

## METHODS

- 
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.ClassDeclaration node )  
 – **Usage**  
   \* Visits a ClassDeclaration  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object **visit**( koala.dynamicjava.tree.ConstructorDeclaration node )  
 – **Usage**  
   \* Visits a ConstructorDeclaration  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object **visit**( koala.dynamicjava.tree.FieldDeclaration node )  
 – **Usage**  
   \* Visits a FieldDeclaration  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object **visit**( koala.dynamicjava.tree.InterfaceDeclaration node )  
 – **Usage**  
   \* Visits a ClassDeclaration  
 – **Parameters**  
   \* node - the node to visit

---

  - *visit*  
 public Object **visit**( koala.dynamicjava.tree.MethodDeclaration node )  
 – **Usage**  
   \* Visits a MethodDeclaration  
 – **Parameters**  
   \* node - the node to visit

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.visitor.VisitorObject

( in 5.2.1, page 160)

- 
- *visit*  
 public Object **visit**( koala.dynamicjava.tree.AddAssignExpression node )  
 – **Usage**  
   \* Visits an AddAssignExpression

- **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

  - **Usage**
  - \* Visits a `AddExpression`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

  - **Usage**
  - \* Visits a `AndExpression`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAccess node )
```

  - **Usage**
  - \* Visits a `ArrayAccess`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

  - **Usage**
  - \* Visits an `ArrayAllocation`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayInitializer node )
```

  - **Usage**
  - \* Visits an `ArrayInitializer`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayType node )
```

  - **Usage**
  - \* Visits a `ArrayType`
  - **Parameters**
  - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )
```

  - **Usage**
  - \* Visits a `BitAndAssignExpression`
  - **Parameters**
  - \* `node` - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.BitAndExpression node )  
 – **Usage**  
   \* Visits a BitAndExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )  
 – **Usage**  
   \* Visits a BitOrAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )  
 – **Usage**  
   \* Visits a BitOrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )  
 – **Usage**  
   \* Visits a BlockStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.BreakStatement node )  
 – **Usage**  
   \* Visits a BreakStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.CastExpression node )  
 – **Usage**  
   \* Visits a CastExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.CatchStatement node )  
 – **Usage**  
   \* Visits a CatchStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassAllocation node )



- **Usage**
    - \* Visits an ClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassDeclaration node )
```

  - **Usage**
    - \* Visits a ClassDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ClassInitializer node )
```

  - **Usage**
    - \* Visits a ClassInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ComplementExpression node )
```

  - **Usage**
    - \* Visits a ComplementExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConditionalExpression node )
```

  - **Usage**
    - \* Visits a ConditionalExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )
```

  - **Usage**
    - \* Visits a ConstructorDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )
```

  - **Usage**
    - \* Visits a ConstructorInvocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ContinueStatement node )
```

  - **Usage**
    - \* Visits a ContinueStatement
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )
      - Usage
        - \* Visits an DivideAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.DivideExpression node )
      - Usage
        - \* Visits a DivideExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.DoStatement node )
      - Usage
        - \* Visits a DoStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.EmptyStatement node )
      - Usage
        - \* Visits an EmptyStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.EqualExpression node )
      - Usage
        - \* Visits a EqualExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
      - Usage
        - \* Visits a ExclusiveOrAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
      - Usage
        - \* Visits a ExclusiveOrExpression
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.FieldDeclaration node )  
 – **Usage**  
   \* Visits a FieldDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FormalParameter node )  
 – **Usage**  
   \* Visits a FormalParameter  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ForStatement node )  
 – **Usage**  
   \* Visits a ForStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.FunctionCall node )  
 – **Usage**  
   \* Visits a FunctionCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterExpression node )  
 – **Usage**  
   \* Visits a GreaterExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )  
 – **Usage**  
   \* Visits a GreaterOrEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )  
 – **Usage**  
   \* Visits a IfThenElseStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.IfThenStatement node )

- **Usage**
    - \* Visits a IfThenStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ImportDeclaration node )
```

  - **Usage**
    - \* Visits an ImportDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerAllocation node )
```

  - **Usage**
    - \* Visits an InnerAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )
```

  - **Usage**
    - \* Visits an InnerClassAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceInitializer node )
```

  - **Usage**
    - \* Visits a InstanceInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )
```

  - **Usage**
    - \* Visits an InstanceOfExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )
```

  - **Usage**
    - \* Visits a InterfaceDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.LabeledStatement node )
```

  - **Usage**
    - \* Visits a LabeledStatement
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
public Object visit( koala.dynamicjava.tree.LessExpression node )
      - Usage
        - \* Visits a LessExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
      - Usage
        - \* Visits a LessOrEqualExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.Literal node )
      - Usage
        - \* Visits a Literal
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
      - Usage
        - \* Visits a MethodDeclaration
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.MinusExpression node )
      - Usage
        - \* Visits a MinusExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
      - Usage
        - \* Visits an MultiplyAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
      - Usage
        - \* Visits a MultiplyExpression
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.NotEqualExpression node )  
 – **Usage**  
   \* Visits a NotEqualExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.NotExpression node )  
 – **Usage**  
   \* Visits a NotExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )  
 – **Usage**  
   \* Visits a ObjectFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )  
 – **Usage**  
   \* Visits a ObjectMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.OrExpression node )  
 – **Usage**  
   \* Visits a OrExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PackageDeclaration node )  
 – **Usage**  
   \* Visits an PackageDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PlusExpression node )  
 – **Usage**  
   \* Visits a PlusExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PostDecrement node )

- **Usage**
    - \* Visits a PostDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PostIncrement node )
```

  - **Usage**
    - \* Visits a PostIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreDecrement node )
```

  - **Usage**
    - \* Visits a PreDecrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PreIncrement node )
```

  - **Usage**
    - \* Visits a PreIncrement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
```

  - **Usage**
    - \* Visits a PrimitiveType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.QualifiedName node )
```

  - **Usage**
    - \* Visits a QualifiedName
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReferenceType node )
```

  - **Usage**
    - \* Visits a ReferenceType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )
```

  - **Usage**
    - \* Visits an RemainderAssignExpression
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*
    - public Object visit( koala.dynamicjava.tree.RemainderExpression node )
    - Usage
      - \* Visits a RemainderExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ReturnStatement node )
    - Usage
      - \* Visits a ReturnStatement
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression node )
    - Usage
      - \* Visits an ShiftLeftAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftLeftExpression node )
    - Usage
      - \* Visits a ShiftLeftExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression node )
    - Usage
      - \* Visits an ShiftRightAssignExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.ShiftRightExpression node )
    - Usage
      - \* Visits a ShiftRightExpression
      - Parameters
      - \* node - the node to visit

---
  - *visit*
    - public Object visit( koala.dynamicjava.tree.SimpleAllocation node )
    - Usage
      - \* Visits an SimpleAllocation
      - Parameters
      - \* node - the node to visit

---



- *visit*  
 public Object visit( koala.dynamicjava.tree.SimpleAssignExpression node )  
 – **Usage**  
   \* Visits an SimpleAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )  
 – **Usage**  
   \* Visits a StaticFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.StaticMethodCall node )  
 – **Usage**  
   \* Visits a StaticMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )  
 – **Usage**  
   \* Visits an SubtractAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractExpression node )  
 – **Usage**  
   \* Visits a SubtractExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )  
 – **Usage**  
   \* Visits a SuperFieldAccess  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SuperMethodCall node )  
 – **Usage**  
   \* Visits a SuperMethodCall  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SwitchBlock node )  


---

- **Usage**
    - \* Visits a SwitchBlock
  - **Parameters**
    - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.SwitchStatement  node )
```

    - **Usage**
      - \* Visits a SwitchStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.SynchronizedStatement  node )
```

    - **Usage**
      - \* Visits a SynchronizedStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThisExpression  node )
```

    - **Usage**
      - \* Visits a ThisExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.ThrowStatement  node )
```

    - **Usage**
      - \* Visits a ThrowStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.TryStatement  node )
```

    - **Usage**
      - \* Visits a TryStatement
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.TypeExpression  node )
```

    - **Usage**
      - \* Visits a TypeExpression
    - **Parameters**
      - \* `node` - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression  node )
```

    - **Usage**
      - \* Visits an UnsignedShiftRightAssignExpression

- **Parameters**
    - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression  node )
```

    - **Usage**
      - \* Visits a UnsignedShiftRightExpression
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration  node )
```

    - **Usage**
      - \* Visits a VariableDeclaration
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.WhileStatement  node )
```

    - **Usage**
      - \* Visits a WhileStatement
    - **Parameters**
      - \* **node** - the node to visit

## 12.2.8 CLASS TreeConstructorInfo

---

The instances of this class provides informations about class constructors not yet compiled to JVM bytecode.

### DECLARATION

---

```
public class TreeConstructorInfo
extends java.lang.Object
implements ConstructorInfo
```

### FIELDS

---

- private ConstructorDeclaration constructorTree
  - The abstract syntax tree of this constructor
- private ClassFinder classFinder
  - The class finder for this class
- private ClassInfo parameters
  - The parameters types
- private ClassInfo exceptions

- The exception types
- private `ClassInfo` `declaringClass`
  - The declaring class
- private `TypeVisitor` `typeVisitor`
  - A visitor to load type infos

## CONSTRUCTORS

---

- *TreeConstructorInfo*

```
public TreeConstructorInfo( koala.dynamicjava.tree.ConstructorDeclaration
f, koala.dynamicjava.classinfo.ClassFinder  cf,
koala.dynamicjava.classinfo.ClassInfo  dc )
```

  - **Usage**
    - \* Creates a new class info
  - **Parameters**
    - \* `f` - the constructor tree
    - \* `cf` - the class finder
    - \* `dc` - the declaring class

## METHODS

---

- *equals*

```
public boolean equals( java.lang.Object  obj )
```

  - **Usage**
    - \* Indicates whether some other object is "equal to" this one
- *getConstructorDeclaration*

```
public ConstructorDeclaration getConstructorDeclaration( )
```

  - **Usage**
    - \* Returns the constructor declaration
- *getExceptionTypes*

```
public ClassInfo getExceptionTypes( )
```

  - **Usage**
    - \* Returns an array of Class infos that represent the types of the exceptions declared to be thrown by the underlying constructor
- *getModifiers*

```
public int getModifiers( )
```

  - **Usage**
    - \* Returns the modifiers for the constructor represented by this object

- *getParameterTypes*

```
public ClassInfo getParameterTypes( )
```

- **Usage**

- \* Returns an array of class infos that represent the parameter types, in declaration order, of the constructor represented by this object

---

- *lookupClass*

```
private ClassInfo lookupClass( java.lang.String s,  
koala.dynamicjava.classinfo.ClassInfo c )
```

- **Usage**

- \* Looks for a class from its name

- **Parameters**

- \* **s** - the name of the class to find
    - \* **c** - the context

- **Exceptions**

- \* `java.lang.NoClassDefFoundError` - if the class cannot be loaded

## 12.2.9 CLASS TreeFieldInfo

---

The instances of this class provides informations about class fields not yet compiled to JVM bytecode.

### DECLARATION

---

```
public class TreeFieldInfo  
extends java.lang.Object  
implements FieldInfo
```

### FIELDS

---

- private FieldDeclaration fieldTree
  - The abstract syntax tree of this field
- private ClassFinder classFinder
  - The class finder for this class
- private ClassInfo type
  - The type of this field
- private ClassInfo declaringClass
  - The declaring class
- private TypeVisitor typeVisitor
  - A visitor to load type infos

CONSTRUCTORS

---

• *TreeFieldInfo*

```
public TreeFieldInfo( koala.dynamicjava.tree.FieldDeclaration  f,
koala.dynamicjava.classinfo.ClassFinder  cf,
koala.dynamicjava.classinfo.ClassInfo  dc )
```

– **Usage**

\* Creates a new class info

– **Parameters**

\* **f** - the field tree

\* **cf** - the class finder

\* **dc** - the declaring class

METHODS

---

• *equals*

```
public boolean equals( java.lang.Object  obj )
```

– **Usage**

\* Indicates whether some other object is "equal to" this one

• *getFieldDeclaration*

```
public FieldDeclaration getFieldDeclaration( )
```

– **Usage**

\* Returns the field declaration

• *getModifiers*

```
public int getModifiers( )
```

– **Usage**

\* Returns the modifiers for the field represented by this object

• *getName*

```
public String getName( )
```

– **Usage**

\* Returns the fully qualified name of the underlying field

• *getType*

```
public ClassInfo getType( )
```

– **Usage**

\* Returns the type of the underlying field

**12.2.10 CLASS *TreeMethodInfo***

---

The instances of this class provides informations about class methods not yet compiled to JVM bytecode.

DECLARATION

---

```
public class TreeMethodInfo
extends java.lang.Object
implements MethodInfo
```

FIELDS

---

- private MethodDeclaration methodTree
  - The abstract syntax tree of this method
- private ClassFinder classFinder
  - The class finder for this class
- private ClassInfo parameters
  - The parameters types
- private ClassInfo exceptions
  - The exception types
- private ClassInfo declaringClass
  - The declaring class
- private TypeVisitor typeVisitor
  - A visitor to load type infos

CONSTRUCTORS

---

- *TreeMethodInfo*  

```
public TreeMethodInfo( koala.dynamicjava.tree.MethodDeclaration f,
koala.dynamicjava.classinfo.ClassFinder cf,
koala.dynamicjava.classinfo.ClassInfo dc )
```

  - **Usage**
    - \* Creates a new method info
  - **Parameters**
    - \* **f** - the method tree
    - \* **cf** - the class finder
    - \* **dc** - the declaring class

METHODS

---

• *equals*

```
public boolean equals( java.lang.Object  obj )
```

– **Usage**

\* Indicates whether some other object is "equal to" this one

---

• *getExceptionTypes*

```
public ClassInfo getExceptionTypes( )
```

– **Usage**

\* Returns an array of Class infos that represent the types of the exceptions declared to be thrown by the underlying method

---

• *getMethodDeclaration*

```
public MethodDeclaration getMethodDeclaration( )
```

– **Usage**

\* Returns the method declaration

---

• *getModifiers*

```
public int getModifiers( )
```

– **Usage**

\* Returns the modifiers for the method represented by this object

---

• *getName*

```
public String getName( )
```

– **Usage**

\* Returns the name of the underlying method

---

• *getParameterTypes*

```
public ClassInfo getParameterTypes( )
```

– **Usage**

\* Returns an array of class infos that represent the parameter types, in declaration order, of the method represented by this object

---

• *getReturnType*

```
public ClassInfo getReturnType( )
```

– **Usage**

\* Returns a Class object that represents the return type of the method represented by this object

---

• *lookupClass*

```
protected ClassInfo lookupClass( java.lang.String  s,  
koala.dynamicjava.classinfo.ClassInfo  c )
```

– **Usage**



- \* Looks for a class from its name
- **Parameters**
  - \* **s** - the name of the class to find
  - \* **c** - the context
- **Exceptions**
  - \* `java.lang.NoClassDefFoundError` - if the class cannot be loaded

### 12.2.11 CLASS *TypeVisitor*

---

The instances of this class are used to get the `ClassInfo` that match a type node of a syntax tree

#### DECLARATION

---

```
public class TypeVisitor
extends koala.dynamicjava.tree.visitor.VisitorObject
```

#### FIELDS

---

- private `ClassFinder` `classFinder`
  - The class finder for this class
- private `ClassInfo` `context`
  - The context

#### CONSTRUCTORS

---

- *TypeVisitor*

```
public TypeVisitor( koala.dynamicjava.classinfo.ClassFinder  cf,
koala.dynamicjava.classinfo.ClassInfo  ctx )
```

  - **Usage**
    - \* Creates a new type visitor
  - **Parameters**
    - \* **cf** - the class finder
    - \* **ctx** - the context

#### METHODS

---

- *lookupClass*

```
private ClassInfo lookupClass( java.lang.String  s,
koala.dynamicjava.classinfo.ClassInfo  c )
```

  - **Usage**
    - \* Looks for a class from its name

- **Parameters**

- \* **s** - the name of the class to find
- \* **c** - the context

- **Exceptions**

- \* `java.lang.NoClassDefFoundError` - if the class cannot be loaded
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayType node )
```

- **Usage**

- \* Visits a `ArrayType`

- **Parameters**

- \* **node** - the node to visit

- **Returns** - the representation of the visited type

- **Exceptions**

- \* `java.lang.NoClassDefFoundError` - if the class cannot be loaded
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.PrimitiveType node )
```

- **Usage**

- \* Visits a `PrimitiveType`

- **Parameters**

- \* **node** - the node to visit

- **Returns** - the representation of the visited type

---

- *visit*

```
public Object visit( koala.dynamicjava.tree.ReferenceType node )
```

- **Usage**

- \* Visits a `ReferenceType`

- **Parameters**

- \* **node** - the node to visit

- **Returns** - the representation of the visited type

- **Exceptions**

- \* `java.lang.NoClassDefFoundError` - if the class cannot be loaded

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.visitor.VisitorObject`

---

( in 5.2.1, page 160)

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddAssignExpression node )
```

- **Usage**

- \* Visits an `AddAssignExpression`

- **Parameters**

- \* **node** - the node to visit
- 

- *visit*

```
public Object visit( koala.dynamicjava.tree.AddExpression node )
```

- **Usage**
    - \* Visits a AddExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.AndExpression node )
```

  - **Usage**
    - \* Visits a AndExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAccess node )
```

  - **Usage**
    - \* Visits a ArrayAccess
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayAllocation node )
```

  - **Usage**
    - \* Visits an ArrayAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayInitializer node )
```

  - **Usage**
    - \* Visits an ArrayInitializer
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ArrayType node )
```

  - **Usage**
    - \* Visits a ArrayType
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndAssignExpression node )
```

  - **Usage**
    - \* Visits a BitAndAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.BitAndExpression node )
```

  - **Usage**
    - \* Visits a BitAndExpression
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrAssignExpression node )
      - Usage
        - \* Visits a BitOrAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.BitOrExpression node )
      - Usage
        - \* Visits a BitOrExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.BlockStatement node )
      - Usage
        - \* Visits a BlockStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.BreakStatement node )
      - Usage
        - \* Visits a BreakStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.CastExpression node )
      - Usage
        - \* Visits a CastExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.CatchStatement node )
      - Usage
        - \* Visits a CatchStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ClassAllocation node )
      - Usage
        - \* Visits an ClassAllocation
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassDeclaration node )  
 – **Usage**  
   \* Visits a ClassDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ClassInitializer node )  
 – **Usage**  
   \* Visits a ClassInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ComplementExpression node )  
 – **Usage**  
   \* Visits a ComplementExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConditionalExpression node )  
 – **Usage**  
   \* Visits a ConditionalExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorDeclaration node )  
 – **Usage**  
   \* Visits a ConstructorDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ConstructorInvocation node )  
 – **Usage**  
   \* Visits a ConstructorInvocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ContinueStatement node )  
 – **Usage**  
   \* Visits a ContinueStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.DivideAssignExpression node )

- **Usage**
    - \* Visits an DivideAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DivideExpression node )
```

  - **Usage**
    - \* Visits a DivideExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.DoStatement node )
```

  - **Usage**
    - \* Visits a DoStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.EmptyStatement node )
```

  - **Usage**
    - \* Visits an EmptyStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.EqualExpression node )
```

  - **Usage**
    - \* Visits a EqualExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrAssignExpression node )
```

  - **Usage**
    - \* Visits a ExclusiveOrAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ExclusiveOrExpression node )
```

  - **Usage**
    - \* Visits a ExclusiveOrExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.FieldDeclaration node )
```

  - **Usage**
    - \* Visits a FieldDeclaration
  - **Parameters**

- 
- \* node - the node to visit

---

    - *visit*  
public Object visit( koala.dynamicjava.tree.FormalParameter node )
      - Usage
        - \* Visits a FormalParameter
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.ForStatement node )
      - Usage
        - \* Visits a ForStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.FunctionCall node )
      - Usage
        - \* Visits a FunctionCall
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.GreaterExpression node )
      - Usage
        - \* Visits a GreaterExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.GreaterOrEqualExpression node )
      - Usage
        - \* Visits a GreaterOrEqualExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.IfThenElseStatement node )
      - Usage
        - \* Visits a IfThenElseStatement
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
public Object visit( koala.dynamicjava.tree.IfThenStatement node )
      - Usage
        - \* Visits a IfThenStatement
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.ImportDeclaration node )  
 – **Usage**  
   \* Visits an ImportDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.InnerAllocation node )  
 – **Usage**  
   \* Visits an InnerAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.InnerClassAllocation node )  
 – **Usage**  
   \* Visits an InnerClassAllocation  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceInitializer node )  
 – **Usage**  
   \* Visits a InstanceInitializer  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.InstanceOfExpression node )  
 – **Usage**  
   \* Visits an InstanceOfExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.InterfaceDeclaration node )  
 – **Usage**  
   \* Visits a InterfaceDeclaration  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.LabeledStatement node )  
 – **Usage**  
   \* Visits a LabeledStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.LessExpression node )  


---



- **Usage**
    - \* Visits a LessExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.LessOrEqualExpression node )
```

  - **Usage**
    - \* Visits a LessOrEqualExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.Literal node )
```

  - **Usage**
    - \* Visits a Literal
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MethodDeclaration node )
```

  - **Usage**
    - \* Visits a MethodDeclaration
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MinusExpression node )
```

  - **Usage**
    - \* Visits a MinusExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyAssignExpression node )
```

  - **Usage**
    - \* Visits an MultiplyAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.MultiplyExpression node )
```

  - **Usage**
    - \* Visits a MultiplyExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.NotEqualExpression node )
```

  - **Usage**
    - \* Visits a NotEqualExpression
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.NotExpression node )
      - Usage
        - \* Visits a NotExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectFieldAccess node )
      - Usage
        - \* Visits a ObjectFieldAccess
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.ObjectMethodCall node )
      - Usage
        - \* Visits a ObjectMethodCall
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.OrExpression node )
      - Usage
        - \* Visits a OrExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.PackageDeclaration node )
      - Usage
        - \* Visits an PackageDeclaration
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.PlusExpression node )
      - Usage
        - \* Visits a PlusExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.PostDecrement node )
      - Usage
        - \* Visits a PostDecrement
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.PostIncrement node )  
 – **Usage**  
   \* Visits a PostIncrement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PreDecrement node )  
 – **Usage**  
   \* Visits a PreDecrement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PreIncrement node )  
 – **Usage**  
   \* Visits a PreIncrement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.PrimitiveType node )  
 – **Usage**  
   \* Visits a PrimitiveType  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.QualifiedName node )  
 – **Usage**  
   \* Visits a QualifiedName  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ReferenceType node )  
 – **Usage**  
   \* Visits a ReferenceType  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderAssignExpression node )  
 – **Usage**  
   \* Visits an RemainderAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.RemainderExpression node )

- **Usage**
    - \* Visits a RemainderExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ReturnStatement  node )
```

  - **Usage**
    - \* Visits a ReturnStatement
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftAssignExpression  node )
```

  - **Usage**
    - \* Visits an ShiftLeftAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftLeftExpression  node )
```

  - **Usage**
    - \* Visits a ShiftLeftExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightAssignExpression  node )
```

  - **Usage**
    - \* Visits an ShiftRightAssignExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.ShiftRightExpression  node )
```

  - **Usage**
    - \* Visits a ShiftRightExpression
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAllocation  node )
```

  - **Usage**
    - \* Visits an SimpleAllocation
  - **Parameters**
    - \* `node` - the node to visit

---
- *visit*

```
public Object visit( koala.dynamicjava.tree.SimpleAssignExpression  node )
```

  - **Usage**
    - \* Visits an SimpleAssignExpression
  - **Parameters**

- 
- \* node - the node to visit
  - - *visit*  
 public Object visit( koala.dynamicjava.tree.StaticFieldAccess node )
      - Usage
        - \* Visits a StaticFieldAccess
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.StaticMethodCall node )
      - Usage
        - \* Visits a StaticMethodCall
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractAssignExpression node )
      - Usage
        - \* Visits an SubtractAssignExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.SubtractExpression node )
      - Usage
        - \* Visits a SubtractExpression
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.SuperFieldAccess node )
      - Usage
        - \* Visits a SuperFieldAccess
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.SuperMethodCall node )
      - Usage
        - \* Visits a SuperMethodCall
      - Parameters
        - \* node - the node to visit

---
  - *visit*  
 public Object visit( koala.dynamicjava.tree.SwitchBlock node )
      - Usage
        - \* Visits a SwitchBlock
      - Parameters
        - \* node - the node to visit

---

- *visit*  
 public Object visit( koala.dynamicjava.tree.SwitchStatement node )  
 – **Usage**  
   \* Visits a SwitchStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.SynchronizedStatement node )  
 – **Usage**  
   \* Visits a SynchronizedStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ThisExpression node )  
 – **Usage**  
   \* Visits a ThisExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.ThrowStatement node )  
 – **Usage**  
   \* Visits a ThrowStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.TryStatement node )  
 – **Usage**  
   \* Visits a TryStatement  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.TypeExpression node )  
 – **Usage**  
   \* Visits a TypeExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightAssignExpression node )  
 – **Usage**  
   \* Visits an UnsignedShiftRightAssignExpression  
 – **Parameters**  
   \* node - the node to visit  


---
- *visit*  
 public Object visit( koala.dynamicjava.tree.UnsignedShiftRightExpression node )

- **Usage**
    - \* Visits a UnsignedShiftRightExpression
  - **Parameters**
    - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.VariableDeclaration  node )
```

    - **Usage**
      - \* Visits a VariableDeclaration
    - **Parameters**
      - \* **node** - the node to visit
- 
- *visit*

```
public Object visit( koala.dynamicjava.tree.WhileStatement  node )
```

    - **Usage**
      - \* Visits a WhileStatement
    - **Parameters**
      - \* **node** - the node to visit

## Chapter 13

# Package koala.dynamicjava.gui

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Interfaces</b>	
<b>MessageHandler</b> .....	802
<i>This interface represents objets that display messages</i>	
<b>Classes</b>	
<b>Editor</b> .....	803
<i>The editor component of the GUI</i>	
<b>Editor.DocumentAdapter</b> .....	831
<i>To listen to the document changes</i>	
<b>Editor.OpenAction</b> .....	832
<i>To open a file</i>	
<b>Editor.RedoAction</b> .....	833
<i>To redo the last undone edit</i>	
<b>Editor.SaveAction</b> .....	834
<i>To save the buffer</i>	
<b>Editor.SaveAsAction</b> .....	835
<i>To save the buffer as a file</i>	
<b>Editor.UndoAction</b> .....	836
<i>To undo the last edit</i>	
<b>Editor.UndoHandler</b> .....	837
<i>To listen to the document undoable edit</i>	
<b>JTextComponentOutputStream</b> .....	838
<i>An output stream that writes to a swing JTextComponent</i>	
<b>Main</b> .....	839
<i>A Graphical User Interface for DynamicJava</i>	
<b>Main.AboutAction</b> .....	864
<i>Pop the About dialog</i>	
<b>Main.ClearAction</b> .....	865
<i>To clear the output</i>	
<b>Main.EditorCaretListener</b> .....	866
<i>To listen to the editor caret</i>	
<b>Main.EvalAction</b> .....	866
<i>To evaluate the content of the buffer</i>	
<b>Main.EvalSelectionAction</b> .....	868
<i>To evaluate the content of the selection</i>	



<b>Main.ExitAction</b> .....	869
<i>To exit the application</i>	
<b>Main.InterpreterThread</b> .....	870
<i>To run the interpreter</i>	
<b>Main.OptionsAction</b> .....	872
<i>To pop the Options dialog</i>	
<b>Main.ReinitAction</b> .....	873
<i>Reinitializes the interpreter</i>	
<b>Main.ScrollBarModelChangeListener</b> .....	874
<i>To listen to the changes in the output area vertical scroll bar model</i>	
<b>Main.StopAction</b> .....	875
<i>To stop the interpreter thread</i>	
<b>OptionsDialog</b> .....	876
<i>The 'options' dialog</i>	
<b>OptionsDialog.CancelButtonAction</b> .....	900
<i>The action associated with the 'Cancel' button</i>	
<b>OptionsDialog.CPLAddButtonAction</b> .....	901
<i>The action associated with the 'add' button of the class path panel</i>	
<b>OptionsDialog.GUIPanel</b> .....	902
<i>The GUI option panel</i>	
<b>OptionsDialog.GUIPanel.CheckBoxChangeListener</b> .....	926
<i>To listen to the checkbox</i>	
<b>OptionsDialog.GUIPanel.FileCheckBoxChangeListener</b> .....	927
<i>To listen to the file checkbox</i>	
<b>OptionsDialog.GUIPanel.InitFileBrowseButtonAction</b> .....	927
<i>The action associated with the 'browse' button</i>	
<b>OptionsDialog.InterpreterPanel</b> .....	928
<i>The interpreter option panel</i>	
<b>OptionsDialog.InterpreterPanel.CheckBoxChangeListener</b> .....	953
<i>To listen to the checkbox</i>	
<b>OptionsDialog.InterpreterPanel.FileCheckBoxChangeListener</b> .....	953
<i>To listen to the file checkbox</i>	
<b>OptionsDialog.InterpreterPanel.InitFileBrowseButtonAction</b> .....	954
<i>The action associated with the 'browse' button</i>	
<b>OptionsDialog.LPLAddButtonAction</b> .....	955
<i>The action associated with the 'add' button of the library path panel</i>	
<b>OptionsDialog.OKButtonAction</b> .....	956
<i>The action associated with the 'OK' button</i>	
<b>OptionsDialog.OptionSet</b> .....	957
<i>To save the options</i>	
<b>OptionsDialog.UCOKButtonAction</b> .....	958
<i>The action associated with the 'OK' button of the URL chooser</i>	
<b>StatusBar</b> .....	959
<i>The status bar</i>	
<b>StatusBar.DisplayThread</b> .....	983
<i>To display the main message</i>	
<b>StringList</b> .....	985
<i>This component is used to manipulate a list of strings</i>	
<b>StringList.DownButtonAction</b> .....	1010
<i>The action associated with the 'down' button</i>	

<b>StringList.ListSelectionAdapter</b> .....	1011
<i>To manage selection modifications</i>	
<b>StringList.RemoveButtonAction</b> .....	1011
<i>The action associated with the 'remove' button</i>	
<b>StringList.UpButtonAction</b> .....	1012
<i>The action associated with the 'up' button</i>	
<b>URLChooser</b> .....	1013
<i>A component used to enter an URL or to choose a local file</i>	
<b>URLChooser.BrowseButtonAction</b> .....	1035
<i>The action associated with the 'browse' button</i>	
<b>URLChooser.CancelButtonAction</b> .....	1036
<i>The action associated with the 'Cancel' button of the URL chooser</i>	
<b>URLChooser.ClearButtonAction</b> .....	1037
<i>The action associated with the 'Clear' button of the URL chooser</i>	
<b>URLChooser.DocumentAdapter</b> .....	1038
<i>To listen to the document changes</i>	
<b>URLChooser.OKButtonAction</b> .....	1039
<i>The action associated with the 'OK' button of the URL chooser</i>	

---

## 13.1 Interfaces

### 13.1.1 INTERFACE MessageHandler

---

This interface represents objects that display messages

#### DECLARATION

---

public interface MessageHandler
---------------------------------

#### METHODS

---

- *setMainMessage*  
 public void **setMainMessage**( java.lang.String s )  
 – **Usage**  
   \* Sets the main message to display  
 – **Parameters**  
   \* s - the message  


---
- *setMainMessage*  
 public void **setMainMessage**( java.lang.String s, java.lang.String s2 )  
 – **Usage**  
   \* Sets the main message to display  
 – **Parameters**  
   \* s - the message  
   \* s2 - a string to concatenate with the message  


---
- *setMessage*  
 public void **setMessage**( java.lang.String s )  
 – **Usage**  
   \* Sets a temporary message to display  
 – **Parameters**  
   \* s - the message  


---
- *setMessage*  
 public void **setMessage**( java.lang.String s, java.lang.String s2 )  
 – **Usage**  
   \* Sets a temporary message to display  
 – **Parameters**  
   \* s - the message  
   \* s2 - a string to concatenate with the message

## 13.2 Classes

### 13.2.1 CLASS Editor

---

The editor component of the GUI

#### DECLARATION

---

```
public class Editor
extends javax.swing.JTextArea
implements koala.dynamicjava.gui.resource.ActionMap
```

#### CONSTRUCTORS

---

- *Editor*  
`public Editor( koala.dynamicjava.gui.MessageHandler mh )`
  - **Usage**
    - \* Creates a new editor
  - **Parameters**
    - \* `mh` - the object that displays the messages

#### METHODS

---

- *closeProcedure*  
`public void closeProcedure( )`
  - **Usage**
    - \* Manages the closing of the buffer
- *getAction*  
`public Action getAction( java.lang.String key )`
  - **Usage**
    - \* Returns the action associated with the given string or null on error
  - **Parameters**
    - \* `key` - the key mapped with the action to get
  - **Exceptions**
    - \* `koala.dynamicjava.gui.resource.MissingListenerException` - if the action is not found
- *openFile*  
`public void openFile( java.lang.String name )`
  - **Usage**
    - \* Opens a file
  - **Parameters**

\* **name** - the name of the file

- *saveDocument*

**protected void saveDocument( )**

– **Usage**

\* Saves the document

#### METHODS INHERITED FROM CLASS javax.swing.JTextArea

---

- *append*

**public void append( java.lang.String )**

- *createDefaultModel*

**protected Document createDefaultModel( )**

- *getAccessibleContext*

**public AccessibleContext getAccessibleContext( )**

- *getColumns*

**public int getColumns( )**

- *getColumnWidth*

**protected int getColumnWidth( )**

- *getLineCount*

**public int getLineCount( )**

- *getLineEndOffset*

**public int getLineEndOffset( int )**

- *getLineOfOffset*

**public int getLineOfOffset( int )**

- *getLineStartOffset*

**public int getLineStartOffset( int )**

- *getLineWrap*

**public boolean getLineWrap( )**

- *getPreferredSizeableViewportSize*

**public Dimension getPreferredSizeableViewportSize( )**

- *getPreferredSize*

**public Dimension getPreferredSize( )**

- *getRowHeight*

**protected int getRowHeight( )**

- *getRows*

**public int getRows( )**

- *getScrollableTracksViewportWidth*

**public boolean getScrollableTracksViewportWidth( )**

- *getScrollableUnitIncrement*

**public int getScrollableUnitIncrement( java.awt.Rectangle , int , int )**

- *getTabSize*

**public int getTabSize( )**

- *getUIClassID*

**public String getUIClassID( )**

- *getWrapStyleWord*

**public boolean getWrapStyleWord( )**

- *insert*

**public void insert( java.lang.String , int )**

- *paramString*  
protected String paramString( )
- *replaceRange*  
public void replaceRange( java.lang.String , int , int )
- *setColumns*  
public void setColumns( int )
- *setFont*  
public void setFont( java.awt.Font )
- *setLineWrap*  
public void setLineWrap( boolean )
- *setRows*  
public void setRows( int )
- *setTabSize*  
public void setTabSize( int )
- *setWrapStyleWord*  
public void setWrapStyleWord( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

---

#### METHODS INHERITED FROM CLASS javax.swing.text.JTextComponent

---

- *<clinit>*  
static void <clinit>( )
- *addCaretListener*  
public void addCaretListener( javax.swing.event.CaretListener )
- *addInputMethodListener*  
public void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeymap*  
public static Keymap addKeymap( java.lang.String , javax.swing.text.Keymap )
- *composedTextExists*  
boolean composedTextExists( )
- *copy*  
public void copy( )
- *createComposedString*  
private void createComposedString( int , java.text.AttributedCharacterIterator )
- *cut*  
public void cut( )
- *exchangeCaret*  
private void exchangeCaret( javax.swing.text.Caret , javax.swing.text.Caret )
- *fireCaretUpdate*  
protected void fireCaretUpdate( javax.swing.event.CaretEvent )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActions*  
public Action getActions( )
- *getCaret*  
public Caret getCaret( )
- *getCaretColor*  
public Color getCaretColor( )

- *getCaretListeners*  
public CaretListener **getCaretListeners**( )
- *getCaretPosition*  
public int **getCaretPosition**( )
- *getCurrentEventModifiers*  
private int **getCurrentEventModifiers**( )
- *getDisabledTextColor*  
public Color **getDisabledTextColor**( )
- *getDocument*  
public Document **getDocument**( )
- *getDragEnabled*  
public boolean **getDragEnabled**( )
- *getFocusAccelerator*  
public char **getFocusAccelerator**( )
- *getFocusedComponent*  
static final JTextComponent **getFocusedComponent**( )
- *getHighlighter*  
public Highlighter **getHighlighter**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeymap*  
public Keymap **getKeymap**( )
- *getKeymap*  
public static Keymap **getKeymap**( java.lang.String )
- *getMargin*  
public Insets **getMargin**( )
- *getNavigationFilter*  
public NavigationFilter **getNavigationFilter**( )
- *getPreferredScrollableViewportSize*  
public Dimension **getPreferredScrollableViewportSize**( )
- *getScrollableBlockIncrement*  
public int **getScrollableBlockIncrement**( java.awt.Rectangle , int , int )
- *getScrollableTracksViewportHeight*  
public boolean **getScrollableTracksViewportHeight**( )
- *getScrollableTracksViewportWidth*  
public boolean **getScrollableTracksViewportWidth**( )
- *getScrollableUnitIncrement*  
public int **getScrollableUnitIncrement**( java.awt.Rectangle , int , int )
- *getSelectedText*  
public String **getSelectedText**( )
- *getSelectedTextColor*  
public Color **getSelectedTextColor**( )
- *getSelectionColor*  
public Color **getSelectionColor**( )
- *getSelectionEnd*  
public int **getSelectionEnd**( )
- *getSelectionStart*  
public int **getSelectionStart**( )
- *getText*  
public String **getText**( )

- *getText*  
public String **getText**( int , int )
- *getToolTipText*  
public String **getToolTipText**( java.awt.event.MouseEvent )
- *getUI*  
public TextUI **getUI**( )
- *installDefaultTransferHandlerIfNecessary*  
private void **installDefaultTransferHandlerIfNecessary**( )
- *invokeAction*  
private void **invokeAction**( java.lang.String , javax.swing.Action )
- *isEditable*  
public boolean **isEditable**( )
- *isProcessInputMethodEventOverridden*  
private boolean **isProcessInputMethodEventOverridden**( )
- *isProcessInputMethodEventOverridden*  
private static Boolean **isProcessInputMethodEventOverridden**( java.lang.Class )
- *loadKeymap*  
public static void **loadKeymap**( javax.swing.text.Keymap ,  
javax.swing.text.JTextComponent.KeyBinding [] , javax.swing.Action [] )
- *mapCommittedTextToAction*  
private void **mapCommittedTextToAction**( java.lang.String )
- *modelToView*  
public Rectangle **modelToView**( int )
- *moveCaretPosition*  
public void **moveCaretPosition**( int )
- *paramString*  
protected String **paramString**( )
- *paste*  
public void **paste**( )
- *processInputMethodEvent*  
protected void **processInputMethodEvent**( java.awt.event.InputMethodEvent )
- *read*  
public void **read**( java.io.Reader , java.lang.Object )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *removeCaretListener*  
public void **removeCaretListener**( javax.swing.event.CaretListener )
- *removeKeymap*  
public static Keymap **removeKeymap**( java.lang.String )
- *removeNotify*  
public void **removeNotify**( )
- *replaceInputMethodText*  
private void **replaceInputMethodText**( java.awt.event.InputMethodEvent )
- *replaceSelection*  
public void **replaceSelection**( java.lang.String )
- *select*  
public void **select**( int , int )
- *selectAll*  
public void **selectAll**( )
- *setCaret*  
public void **setCaret**( javax.swing.text.Caret )



- *setCaretColor*  
public void setCaretColor( java.awt.Color    )
- *setCaretPosition*  
public void setCaretPosition( int    )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation    )
- *setDisabledTextColor*  
public void setDisabledTextColor( java.awt.Color    )
- *setDocument*  
public void setDocument( javax.swing.text.Document    )
- *setDragEnabled*  
public void setDragEnabled( boolean    )
- *setEditable*  
public void setEditable( boolean    )
- *setFocusAccelerator*  
public void setFocusAccelerator( char    )
- *setHighlighter*  
public void setHighlighter( javax.swing.text.Highlighter    )
- *setInputMethodCaretPosition*  
private void setInputMethodCaretPosition( java.awt.event.InputMethodEvent    )
- *setKeymap*  
public void setKeymap( javax.swing.text.Keymap    )
- *setMargin*  
public void setMargin( java.awt.Insets    )
- *setNavigationFilter*  
public void setNavigationFilter( javax.swing.text.NavigationFilter    )
- *setSelectedTextColor*  
public void setSelectedTextColor( java.awt.Color    )
- *setSelectionColor*  
public void setSelectionColor( java.awt.Color    )
- *setSelectionEnd*  
public void setSelectionEnd( int    )
- *setSelectionStart*  
public void setSelectionStart( int    )
- *setText*  
public void setText( java.lang.String    )
- *setUI*  
public void setUI( javax.swing.plaf.TextUI    )
- *shouldSynthesizeKeyEvents*  
private boolean shouldSynthesizeKeyEvents(    )
- *updateInputMap*  
void updateInputMap( javax.swing.text.Keymap    , javax.swing.text.Keymap    )
- *updateUI*  
public void updateUI(    )
- *viewToModel*  
public int viewToModel( java.awt.Point    )
- *write*  
public void write( java.io.Writer    )

## METHODS INHERITED FROM CLASS javax.swing.JComponent

- 
- *\_paintImmediately*  
void **\_paintImmediately**( int , int , int , int )
  - *<clinit>*  
static void **<clinit>**( )
  - *addAncestorListener*  
public void **addAncestorListener**( javax.swing.event.AncestorListener )
  - *addNotify*  
public void **addNotify**( )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.beans.PropertyChangeListener )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.lang.String , java.beans.PropertyChangeListener )
  - *addVetoableChangeListener*  
public synchronized void **addVetoableChangeListener**( java.beans.VetoableChangeListener )
  - *adjustPaintFlags*  
private void **adjustPaintFlags**( )
  - *alwaysOnTop*  
boolean **alwaysOnTop**( )
  - *checkIfChildObscuredBySibling*  
boolean **checkIfChildObscuredBySibling**( )
  - *componentInputMapChanged*  
void **componentInputMapChanged**( javax.swing.ComponentInputMap )
  - *computeVisibleRect*  
static final void **computeVisibleRect**( java.awt.Component , java.awt.Rectangle )
  - *computeVisibleRect*  
public void **computeVisibleRect**( java.awt.Rectangle )
  - *compWriteObjectNotify*  
void **compWriteObjectNotify**( )
  - *contains*  
public boolean **contains**( int , int )
  - *createToolTip*  
public JToolTip **createToolTip**( )
  - *deregisterNextFocusableComponent*  
private void **deregisterNextFocusableComponent**( )
  - *disable*  
public void **disable**( )
  - *enable*  
public void **enable**( )
  - *enableSerialization*  
void **enableSerialization**( )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , boolean , boolean )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , byte , byte )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , char , char )

- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getAncestorListeners*  
public AncestorListener getAncestorListeners( )
- *getAutoscrolls*  
public boolean getAutoscrolls( )
- *getBorder*  
public Border getBorder( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary getClientProperties( )
- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )
- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )

- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )

- *getToolTipText*  
public String **getToolTipText**( )
- *getToolTipText*  
public String **getToolTipText**( java.awt.event.MouseEvent    )
- *getTopLevelAncestor*  
public Container **getTopLevelAncestor**( )
- *getTransferHandler*  
public TransferHandler **getTransferHandler**( )
- *getUIClassID*  
public String **getUIClassID**( )
- *getVerifyInputWhenFocusTarget*  
public boolean **getVerifyInputWhenFocusTarget**( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener **getVetoableChangeListeners**( )
- *getWidth*  
public int **getWidth**( )
- *getVisibleRect*  
public Rectangle **getVisibleRect**( )
- *getWriteObjCounter*  
static byte **getWriteObjCounter**( javax.swing.JComponent    )
- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *grabFocus*  
public void **grabFocus**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isLightweightComponent*  
public static boolean **isLightweightComponent**( java.awt.Component    )
- *isManagingFocus*  
public boolean **isManagingFocus**( )
- *isMaximumSizeSet*  
public boolean **isMaximumSizeSet**( )
- *isMinimumSizeSet*  
public boolean **isMinimumSizeSet**( )
- *isOpaque*  
public boolean **isOpaque**( )
- *isOptimizedDrawingEnabled*  
public boolean **isOptimizedDrawingEnabled**( )
- *isPaintingOrigin*  
boolean **isPaintingOrigin**( )
- *isPaintingTile*  
public boolean **isPaintingTile**( )
- *isPreferredSizeSet*  
public boolean **isPreferredSizeSet**( )
- *isRequestFocusEnabled*  
public boolean **isRequestFocusEnabled**( )
- *isValidateRoot*  
public boolean **isValidateRoot**( )

- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )

- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener(  
java.beans.VetoableChangeListener )
- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )

- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )



- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )

- *adjustDecendantsOnParent*  
void **adjustDecendantsOnParent**( int )
- *adjustDescendants*  
void **adjustDescendants**( int )
- *adjustListeningChildren*  
void **adjustListeningChildren**( long , int )
- *applyComponentOrientation*  
public void **applyComponentOrientation**( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean **areFocusTraversalKeysSet**( int )
- *checkGD*  
void **checkGD**( java.lang.String )
- *clearCurrentFocusCycleRootOnHide*  
void **clearCurrentFocusCycleRootOnHide**( )
- *clearMostRecentFocusOwnerOnHide*  
void **clearMostRecentFocusOwnerOnHide**( )
- *containsFocus*  
final boolean **containsFocus**( )
- *countComponents*  
public int **countComponents**( )
- *countHierarchyMembers*  
int **countHierarchyMembers**( )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int , long , boolean )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void **deliverEvent**( java.awt.Event )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void **dispatchEventToSelf**( java.awt.AWTEvent )
- *doLayout*  
public void **doLayout**( )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent )
- *findComponentAt*  
public Component **findComponentAt**( int , int )
- *findComponentAt*  
final Component **findComponentAt**( int , int , boolean )
- *findComponentAt*  
public Component **findComponentAt**( java.awt.Point )
- *findTraversalRoot*  
private Container **findTraversalRoot**( )
- *getAccessibleAt*  
Accessible **getAccessibleAt**( java.awt.Point )
- *getAccessibleChild*  
Accessible **getAccessibleChild**( int )
- *getAccessibleChildrenCount*  
int **getAccessibleChildrenCount**( )

- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getComponent*  
public Component **getComponent**( int )
- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentCount*  
public int **getComponentCount**( )
- *getComponents\_NoClientCode*  
final Component **getComponents\_NoClientCode**( )
- *getComponents*  
public Component **getComponents**( )
- *getContainerListeners*  
public synchronized ContainerListener **getContainerListeners**( )
- *getDropTargetEventTarget*  
Component **getDropTargetEventTarget**( int , int , boolean )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy **getFocusTraversalPolicy**( )
- *getInsets*  
public Insets **getInsets**( )
- *getLayout*  
public LayoutManager **getLayout**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )

- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )

- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )
- *removeAll*  
public void **removeAll**(    )
- *removeContainerListener*  
public synchronized void **removeContainerListener**( java.awt.event.ContainerListener    )
- *removeNotify*  
public void **removeNotify**(    )
- *setFocusCycleRoot*  
public void **setFocusCycleRoot**( boolean    )
- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int    , java.util.Set    )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy    )
- *setFont*  
public void **setFont**( java.awt.Font    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setZOrder*  
void **setZOrder**( java.awt.Component    , int    )
- *transferFocusBackward*  
public void **transferFocusBackward**(    )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**(    )
- *update*  
public void **update**( java.awt.Graphics    )
- *validate*  
public void **validate**(    )
- *validateTree*  
protected void **validateTree**(    )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )

METHODS INHERITED FROM CLASS `java.awt.Component`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *action*  
`public boolean action( java.awt.Event , java.lang.Object )`
  - *add*  
`public synchronized void add( java.awt.PopupMenu )`
  - *addComponentListener*  
`public synchronized void addComponentListener( java.awt.event.ComponentListener )`
  - *addFocusListener*  
`public synchronized void addFocusListener( java.awt.event.FocusListener )`
  - *addHierarchyBoundsListener*  
`public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )`
  - *addHierarchyListener*  
`public void addHierarchyListener( java.awt.event.HierarchyListener )`
  - *addInputMethodListener*  
`public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )`
  - *addKeyListener*  
`public synchronized void addKeyListener( java.awt.event.KeyListener )`
  - *addMouseListener*  
`public synchronized void addMouseListener( java.awt.event.MouseListener )`
  - *addMouseMotionListener*  
`public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )`
  - *addMouseWheelListener*  
`public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )`
  - *addNotify*  
`public void addNotify( )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
  - *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
  - *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
  - *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
  - *areInputMethodsEnabled*  
`boolean areInputMethodsEnabled( )`
  - *autoProcessMouseWheel*  
`void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )`
  - *autoTransferFocus*  
`final void autoTransferFocus( boolean )`

- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )

- *dispatchEvent*  
public final void **dispatchEvent**( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean **dispatchMouseWheelToAncestor**( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void **doAutoTransfer**( boolean    )
- *doLayout*  
public void **doLayout**( )
- *enable*  
public void **enable**( )
- *enable*  
public void **enable**( boolean    )
- *enableEvents*  
protected final void **enableEvents**( long    )
- *enableInputMethods*  
public void **enableInputMethods**( boolean    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean **eventTypeEnabled**( int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , int    , int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , java.lang.Object    ,  
java.lang.Object    )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getAccessibleIndexInParent*  
int **getAccessibleIndexInParent**( )
- *getAccessibleStateSet*  
AccessibleStateSet **getAccessibleStateSet**( )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle    )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )



- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )

- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener **getMouseMotionListeners**( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener **getMouseWheelListeners**( )
- *getName*  
public String **getName**( )
- *getNativeContainer*  
Container **getNativeContainer**( )
- *getParent\_NoClientCode*  
final Container **getParent\_NoClientCode**( )
- *getParent*  
public Container **getParent**( )
- *getPeer*  
public ComponentPeer **getPeer**( )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getSize*  
public Dimension **getSize**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getToolkitImpl*  
final Toolkit **getToolkitImpl**( )
- *getTreeLock*  
public final Object **getTreeLock**( )
- *getWidth*  
public int **getWidth**( )

- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *gotFocus*  
public boolean **gotFocus**( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean **handleEvent**( java.awt.Event )
- *hasFocus*  
public boolean **hasFocus**( )
- *hide*  
public void **hide**( )
- *imageUpdate*  
public boolean **imageUpdate**( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *inside*  
public boolean **inside**( int , int )
- *invalidate*  
public void **invalidate**( )
- *isBackgroundSet*  
public boolean **isBackgroundSet**( )
- *isCursorSet*  
public boolean **isCursorSet**( )
- *isDisplayable*  
public boolean **isDisplayable**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isEnabled*  
public boolean **isEnabled**( )
- *isEnabledImpl*  
final boolean **isEnabledImpl**( )
- *isFocusable*  
public boolean **isFocusable**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusOwner*  
public boolean **isFocusOwner**( )
- *isFocusTraversable*  
public boolean **isFocusTraversable**( )
- *isFocusTraversableOverridden*  
final boolean **isFocusTraversableOverridden**( )
- *isFontSet*  
public boolean **isFontSet**( )
- *isForegroundSet*  
public boolean **isForegroundSet**( )
- *isLightweight*  
public boolean **isLightweight**( )

- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )

- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvent*  
boolean postsOldMouseEvent( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )

- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *repaint*  
public void repaint( )
- *repaint*  
public void repaint( int , int , int , int )
- *repaint*  
public void repaint( long )
- *repaint*  
public void repaint( long , int , int , int , int )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )

- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )

- *setSize*  
public void setSize( java.awt.Dimension    )
- *setSize*  
public void setSize( int    , int    )
- *setVisible*  
public void setVisible( boolean    )
- *show*  
public void show(    )
- *show*  
public void show( boolean    )
- *size*  
public Dimension size(    )
- *toString*  
public String toString(    )
- *transferFocus*  
public void transferFocus(    )
- *transferFocusBackward*  
public void transferFocusBackward(    )
- *transferFocusUpCycle*  
public void transferFocusUpCycle(    )
- *update*  
public void update( java.awt.Graphics    )
- *updateCursorImmediately*  
final void updateCursorImmediately(    )
- *validate*  
public void validate(    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

### 13.2.2 CLASS Editor.DocumentAdapter

---

To listen to the document changes

#### DECLARATION

---

```
class Editor.DocumentAdapter
extends java.lang.Object
implements javax.swing.event.DocumentListener
```

#### CONSTRUCTORS

---

- *Editor.DocumentAdapter*  
**Editor.DocumentAdapter(    )**



## METHODS

- *changedUpdate*  
public void **changedUpdate**( javax.swing.event.DocumentEvent e )
- *insertUpdate*  
public void **insertUpdate**( javax.swing.event.DocumentEvent e )
- *removeUpdate*  
public void **removeUpdate**( javax.swing.event.DocumentEvent e )

## 13.2.3 CLASS Editor.OpenAction

To open a file

## DECLARATION

```
class Editor.OpenAction
extends javax.swing.AbstractAction
```

## CONSTRUCTORS

- *Editor.OpenAction*  
**Editor.OpenAction( )**

## METHODS

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

## METHODS INHERITED FROM CLASS javax.swing.AbstractAction

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object **clone**( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object **getKeys**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object **getValue**( java.lang.String )

- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.4 CLASS Editor.RedoAction

---

To redo the last undone edit

#### DECLARATION

---

```
class Editor.RedoAction
extends javax.swing.AbstractAction
implements koala.dynamicjava.gui.resource.JComponentModifier
```

#### CONSTRUCTORS

---

- *Editor.RedoAction*  
**Editor.RedoAction( )**

#### METHODS

---

- *actionPerformed*  
public void actionPerformed( java.awt.event.ActionEvent e )
- *addJComponent*  
public void addJComponent( javax.swing.JComponent c )
- *update*  
protected void update( )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )

- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.5 CLASS Editor.SaveAction

---

To save the buffer

#### DECLARATION

---

```
class Editor.SaveAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *Editor.SaveAction*  
**Editor.SaveAction( )**

#### METHODS

---

- *actionPerformed*  
**public void actionPerformed( java.awt.event.ActionEvent e )**

METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

**13.2.6 CLASS Editor.SaveAsAction**

---

To save the buffer as a file

DECLARATION

---

```
class Editor.SaveAsAction
extends javax.swing.AbstractAction
```

CONSTRUCTORS

---

- *Editor.SaveAsAction*  
**Editor.SaveAsAction**( )

METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *clone*  
protected Object clone(    )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String   , java.lang.Object   ,   
java.lang.Object    )
- *getKeys*  
public Object getKeys(    )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners(    )
- *getValue*  
public Object getValue( java.lang.String    )
- *isEnabled*  
public boolean isEnabled(    )
- *putValue*  
public void putValue( java.lang.String   , java.lang.Object    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *setEnabled*  
public void setEnabled( boolean    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

### 13.2.7 CLASS Editor.UndoAction

---

To undo the last edit

#### DECLARATION

---

```
class Editor.UndoAction
extends javax.swing.AbstractAction
implements koala.dynamicjava.gui.resource.JComponentModifier
```

#### CONSTRUCTORS

---

- *Editor.UndoAction*  
**Editor.UndoAction(    )**

## METHODS

- *actionPerformed*  
public void actionPerformed( java.awt.event.ActionEvent e )
- *addJComponent*  
public void addJComponent( javax.swing.JComponent c )
- *update*  
protected void update( )

## METHODS INHERITED FROM CLASS javax.swing.AbstractAction

- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

## 13.2.8 CLASS Editor.UndoHandler

To listen to the document undoable edit

## DECLARATION

```
class Editor.UndoHandler
extends java.lang.Object
implements javax.swing.event.UndoableEditListener
```

CONSTRUCTORS

---

- *Editor.UndoHandler*  
**Editor.UndoHandler**( )

METHODS

---

- *undoableEditHappened*  
**public void undoableEditHappened**( javax.swing.event.UndoableEditEvent e )
  - **Usage**
    - \* Messaged when the Document has created an edit, the edit is added to **undo**, an instance of UndoManager.

**13.2.9 CLASS JTextComponentOutputStream**

---

An output stream that writes to a swing JTextComponent

DECLARATION

---

```
public class JTextComponentOutputStream
extends java.io.OutputStream
```

CONSTRUCTORS

---

- *JTextComponentOutputStream*  
**public JTextComponentOutputStream**( javax.swing.text.JTextComponent tc )
  - **Usage**
    - \* Creates a new output stream
  - **Parameters**
    - \* **tc** - the text component that will receive the output

METHODS

---

- *write*  
**public void write**( byte [] b, int off, int len )
  - **Usage**

- \* Writes **len** bytes from the specified byte array starting at offset **off** to this output stream.

The **write** method of **OutputStream** calls the write method of one argument on each of the bytes to be written out. Subclasses are encouraged to override this method and provide a more efficient implementation.

– **Parameters**

- \* **b** - the data.
- \* **off** - the start offset in the data.
- \* **len** - the number of bytes to write.

– **Exceptions**

- \* **java.io.IOException** - if an I/O error occurs.

---

• *write*

```
public void write( int  b )
```

– **Usage**

- \* Writes the specified byte to this output stream.

Subclasses of **OutputStream** must provide an implementation for this method.

– **Parameters**

- \* **b** - the byte.

– **Exceptions**

- \* **java.io.IOException** - if an I/O error occurs.

---

#### METHODS INHERITED FROM CLASS **java.io.OutputStream**

• *close*

```
public void close( )
```

• *flush*

```
public void flush( )
```

• *write*

```
public void write( byte []  )
```

• *write*

```
public void write( byte []  , int  , int  )
```

• *write*

```
public abstract void write( int  )
```

---

### 13.2.10 CLASS **Main**

A Graphical User Interface for DynamicJava

---

#### DECLARATION

```
public class Main
extends javax.swing.JFrame
implements koala.dynamicjava.gui.resource.ActionMap
```



FIELDS

---

- public static final String OPEN\_ACTION  
—
- public static final String SAVE\_ACTION  
—
- public static final String SAVE\_AS\_ACTION  
—
- public static final String EXIT\_ACTION  
—
- public static final String UNDO\_ACTION  
—
- public static final String REDO\_ACTION  
—
- public static final String CUT\_ACTION  
—
- public static final String COPY\_ACTION  
—
- public static final String PASTE\_ACTION  
—
- public static final String CLEAR\_ACTION  
—
- public static final String OPTIONS\_ACTION  
—
- public static final String EVAL\_ACTION  
—
- public static final String EVAL\_S\_ACTION  
—
- public static final String STOP\_ACTION  
—
- public static final String REINIT\_ACTION  
—
- public static final String ABOUT\_ACTION  
—

## CONSTRUCTORS

---

- *Main*  
**public Main( )**
  - **Usage**
    - \* Creates the interface

## METHODS

---

- *applyOptions*  
**protected void applyOptions( )**
  - **Usage**
    - \* Applies the options

---
- *createInterpreter*  
**protected Interpreter createInterpreter( )**
  - **Usage**
    - \* Creates a new interpreter

---
- *exit*  
**protected void exit( )**
  - **Usage**
    - \* Called when the interface exits

---
- *getAction*  
**public Action getAction( java.lang.String key )**
  - **Usage**
    - \* Returns the action associated with the given string or null on error
  - **Parameters**
    - \* **key** - the key mapped with the action to get
  - **Exceptions**
    - \* `koala.dynamicjava.gui.resource.MissingListenerException` - if the action is not found

---
- *getCurrentLine*  
**protected int getCurrentLine( int p )**
  - **Usage**
    - \* Returns the line number that match the given position
  - **Parameters**
    - \* **p** - a position

---
- *getEditor*  
**public Editor getEditor( )**

- **Usage**
  - \* Returns the editor

---
- *getOptions*  
 public OptionsDialog.OptionSet **getOptions**( )
  - **Usage**
  - \* Gets the options

---
- *getOptionsDialog*  
 public OptionsDialog **getOptionsDialog**( )
  - **Usage**
  - \* Returns the options dialog

---
- *getOutputArea*  
 public JTextArea **getOutputArea**( )
  - **Usage**
  - \* Returns the output area

---
- *loadOptions*  
 public void **loadOptions**( )
  - **Usage**
  - \* Loads the options

---
- *main*  
 public static void **main**( java.lang.String [] args )
  - **Usage**
  - \* The entry point of the program

---
- *reinitializeInterpreter*  
 protected void **reinitializeInterpreter**( )
  - **Usage**
  - \* Reinitializes the interpreter

---
- *restoreOptions*  
 protected void **restoreOptions**( )
  - **Usage**
  - \* Restores the options

---
- *saveOptions*  
 public void **saveOptions**( )
  - **Usage**
  - \* Saves the options to System.getProperty("user.home") + "/.djava/options"

---

- *setOptions*

```
public void setOptions( koala.dynamicjava.gui.OptionsDialog.OptionSet opt )
```

- Usage

- \* Sets the options

---

- *stringToJavaString*

```
protected String stringToJavaString( java.lang.String s )
```

- Usage

- \* translates a string to a java source string

## METHODS INHERITED FROM CLASS javax.swing.JFrame

---

- *<clinit>*

```
static void <clinit>( )
```

---

- *addImpl*

```
protected void addImpl( java.awt.Component , java.lang.Object , int )
```

---

- *createRootPane*

```
protected JRootPane createRootPane( )
```

---

- *createRootPaneException*

```
private Error createRootPaneException( java.lang.String )
```

---

- *frameInit*

```
protected void frameInit( )
```

---

- *getAccessibleContext*

```
public AccessibleContext getAccessibleContext( )
```

---

- *getContentPane*

```
public Container getContentPane( )
```

---

- *getDefaultCloseOperation*

```
public int getDefaultCloseOperation( )
```

---

- *getGlassPane*

```
public Component getGlassPane( )
```

---

- *getJMenuBar*

```
public JMenuBar getJMenuBar( )
```

---

- *getLayeredPane*

```
public JLayeredPane getLayeredPane( )
```

---

- *getRootPane*

```
public JRootPane getRootPane( )
```

---

- *isDefaultLookAndFeelDecorated*

```
public static boolean isDefaultLookAndFeelDecorated( )
```

---

- *isRootPaneCheckingEnabled*

```
protected boolean isRootPaneCheckingEnabled( )
```

---

- *paramString*

```
protected String paramString( )
```

---

- *processWindowEvent*

```
protected void processWindowEvent( java.awt.event.WindowEvent )
```

---

- *remove*

```
public void remove( java.awt.Component )
```

---

- *setContentPane*  
public void **setContentPane**( java.awt.Container    )
- *setDefaultCloseOperation*  
public void **setDefaultCloseOperation**( int    )
- *setDefaultLookAndFeelDecorated*  
public static void **setDefaultLookAndFeelDecorated**( boolean    )
- *setGlassPane*  
public void **setGlassPane**( java.awt.Component    )
- *setJMenuBar*  
public void **setJMenuBar**( javax.swing.JMenuBar    )
- *setLayeredPane*  
public void **setLayeredPane**( javax.swing.JLayeredPane    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setRootPane*  
protected void **setRootPane**( javax.swing.JRootPane    )
- *setRootPaneCheckingEnabled*  
protected void **setRootPaneCheckingEnabled**( boolean    )
- *update*  
public void **update**( java.awt.Graphics    )

#### METHODS INHERITED FROM CLASS java.awt.Frame

---

- *<clinit>*  
static void **<clinit>**( )
- *addNotify*  
public void **addNotify**( )
- *addToFrameList*  
void **addToFrameList**( )
- *constructComponentName*  
String **constructComponentName**( )
- *finalize*  
protected void **finalize**( )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getCursorType*  
public int **getCursorType**( )
- *getExtendedState*  
public synchronized int **getExtendedState**( )
- *getFrames*  
public static Frame **getFrames**( )
- *getIconImage*  
public Image **getIconImage**( )
- *getMaximizedBounds*  
public Rectangle **getMaximizedBounds**( )
- *getMenuBar*  
public MenuBar **getMenuBar**( )
- *getState*  
public synchronized int **getState**( )

- *getTitle*  
public String getTitle( )
- *init*  
private void init( java.lang.String , java.awt.GraphicsConfiguration )
- *initIDs*  
private static native void initIDs( )
- *isResizable*  
public boolean isResizable( )
- *isUndecorated*  
public boolean isUndecorated( )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public void remove( java.awt.MenuComponent )
- *removeFromFrameList*  
void removeFromFrameList( )
- *removeNotify*  
public void removeNotify( )
- *setCursor*  
public void setCursor( int )
- *setExtendedState*  
public synchronized void setExtendedState( int )
- *setIconImage*  
public synchronized void setIconImage( java.awt.Image )
- *setMaximizedBounds*  
public synchronized void setMaximizedBounds( java.awt.Rectangle )
- *setMenuBar*  
public void setMenuBar( java.awt.MenuBar )
- *setResizable*  
public void setResizable( boolean )
- *setState*  
public synchronized void setState( int )
- *setTitle*  
public void setTitle( java.lang.String )
- *setUndecorated*  
public void setUndecorated( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

METHODS INHERITED FROM CLASS `java.awt.Window`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *addNotify*  
`public void addNotify( )`
  - *addOwnedWindow*  
`void addOwnedWindow( java.lang.ref.WeakReference )`
  - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )`
  - *addWindowFocusListener*  
`public synchronized void addWindowFocusListener(  
java.awt.event.WindowFocusListener )`
  - *addWindowListener*  
`public synchronized void addWindowListener( java.awt.event.WindowListener )`
  - *addWindowStateListener*  
`public synchronized void addWindowStateListener(  
java.awt.event.WindowStateListener )`
  - *adjustDecendantsOnParent*  
`void adjustDecendantsOnParent( int )`
  - *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
  - *applyResourceBundle*  
`public void applyResourceBundle( java.util.ResourceBundle )`
  - *applyResourceBundle*  
`public void applyResourceBundle( java.lang.String )`
  - *clearMostRecentFocusOwnerOnHide*  
`final void clearMostRecentFocusOwnerOnHide( )`
  - *connectOwnedWindow*  
`void connectOwnedWindow( java.awt.Window )`
  - *constructComponentName*  
`String constructComponentName( )`
  - *createBufferStrategy*  
`public void createBufferStrategy( int )`
  - *createBufferStrategy*  
`public void createBufferStrategy( int , java.awt.BufferCapabilities )`
  - *deliverMouseWheelToAncestor*  
`void deliverMouseWheelToAncestor( java.awt.event.MouseWheelEvent )`
  - *dispatchEventImpl*  
`void dispatchEventImpl( java.awt.AWTEvent )`
  - *dispatchMouseWheelToAncestor*  
`boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )`
  - *dispose*  
`public void dispose( )`
  - *eventEnabled*  
`boolean eventEnabled( java.awt.AWTEvent )`
  - *finalize*  
`protected void finalize( )`

- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getBufferStrategy*  
public BufferStrategy getBufferStrategy( )
- *getFocusableWindowState*  
public boolean getFocusableWindowState( )
- *getFocusCycleRootAncestor*  
public final Container getFocusCycleRootAncestor( )
- *getFocusOwner*  
public Component getFocusOwner( )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getInputContext*  
public InputContext getInputContext( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocale*  
public Locale getLocale( )
- *getMostRecentFocusOwner*  
public Component getMostRecentFocusOwner( )
- *getOwnedWindows*  
public Window getOwnedWindows( )
- *getOwner*  
public Window getOwner( )
- *getTemporaryLostComponent*  
Component getTemporaryLostComponent( )
- *getToolkit*  
public Toolkit getToolkit( )
- *getWarningString*  
public final String getWarningString( )
- *getWindowFocusListeners*  
public synchronized WindowFocusListener getWindowFocusListeners( )
- *getWindowListeners*  
public synchronized WindowListener getWindowListeners( )
- *getWindowStateListeners*  
public synchronized WindowStateListener getWindowStateListeners( )
- *hide*  
public void hide( )
- *init*  
private void init( java.awt.GraphicsConfiguration )
- *initIDs*  
private static native void initIDs( )
- *isActive*  
public boolean isActive( )
- *isFocusableWindow*  
public final boolean isFocusableWindow( )
- *isFocusCycleRoot*  
public final boolean isFocusCycleRoot( )



- *isFocused*  
public boolean isFocused( )
- *isShowing*  
public boolean isShowing( )
- *ownedInit*  
private void ownedInit( java.awt.Window )
- *pack*  
public void pack( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postWindowEvent*  
synchronized void postWindowEvent( int )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processWindowEvent*  
protected void processWindowEvent( java.awt.event.WindowEvent )
- *processWindowFocusEvent*  
protected void processWindowFocusEvent( java.awt.event.WindowEvent )
- *processWindowStateEvent*  
protected void processWindowStateEvent( java.awt.event.WindowEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removeOwnedWindow*  
void removeOwnedWindow( java.lang.ref.WeakReference )
- *removeWindowFocusListener*  
public synchronized void removeWindowFocusListener( java.awt.event.WindowFocusListener )
- *removeWindowListener*  
public synchronized void removeWindowListener( java.awt.event.WindowListener )
- *removeWindowStateListener*  
public synchronized void removeWindowStateListener( java.awt.event.WindowStateListener )
- *resetGC*  
void resetGC( )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setFocusableWindowState*  
public void setFocusableWindowState( boolean )
- *setFocusCycleRoot*  
public final void setFocusCycleRoot( boolean )
- *setLocationRelativeTo*  
public void setLocationRelativeTo( java.awt.Component )
- *setTemporaryLostComponent*  
Component setTemporaryLostComponent( java.awt.Component )
- *setWarningString*  
private void setWarningString( )

- *show*  
public void show( )
- *toBack*  
public void toBack( )
- *toFront*  
public void toFront( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int )
- *adjustDescendants*  
void adjustDescendants( int )
- *adjustListeningChildren*  
void adjustListeningChildren( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *checkGD*  
void checkGD( java.lang.String )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )

- *containsFocus*  
final boolean containsFocus( )
- *countComponents*  
public int countComponents( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )

- *getContainerListeners*  
public synchronized ContainerListener **getContainerListeners**( )
- *getDropTargetEventTarget*  
Component **getDropTargetEventTarget**( int , int , boolean )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy **getFocusTraversalPolicy**( )
- *getInsets*  
public Insets **getInsets**( )
- *getLayout*  
public LayoutManager **getLayout**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )
- *invalidate*  
public void **invalidate**( )
- *invalidateTree*  
void **invalidateTree**( )
- *isAncestorOf*  
public boolean **isAncestorOf**( java.awt.Component )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean **isFocusTraversalPolicySet**( )
- *isParentOf*  
boolean **isParentOf**( java.awt.Component )
- *layout*  
public void **layout**( )

- *lightweightPaint*  
void **lightweightPaint**( java.awt.Graphics )
- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics )
- *list*  
public void **list**( java.io.PrintStream , int )
- *list*  
public void **list**( java.io.PrintWriter , int )
- *locate*  
public Component **locate**( int , int )
- *minimumSize*  
public Dimension **minimumSize**( )
- *nextFocusHelper*  
boolean **nextFocusHelper**( )
- *numListening*  
int **numListening**( long )
- *paint*  
public void **paint**( java.awt.Graphics )
- *paintComponents*  
public void **paintComponents**( java.awt.Graphics )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics )
- *paramString*  
protected String **paramString**( )
- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *preProcessKeyEvent*  
void **preProcessKeyEvent**( java.awt.event.KeyEvent )
- *print*  
public void **print**( java.awt.Graphics )
- *printComponents*  
public void **printComponents**( java.awt.Graphics )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *remove*  
public void **remove**( java.awt.Component )
- *remove*  
public void **remove**( int )

- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void setFocusTraversalPolicy( java.awt.FocusTraversalPolicy )
- *setFont*  
public void setFont( java.awt.Font )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setZOrder*  
void setZOrder( java.awt.Component , int )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusDownCycle*  
public void transferFocusDownCycle( )
- *update*  
public void update( java.awt.Graphics )
- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )

- *addInputMethodListener*  
public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )
- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *areInputMethodsEnabled*  
boolean areInputMethodsEnabled( )
- *autoProcessMouseWheel*  
void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )
- *autoTransferFocus*  
final void autoTransferFocus( boolean )
- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )

- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *doAutoTransfer*  
private void doAutoTransfer( boolean )
- *doLayout*  
public void doLayout( )
- *enable*  
public void enable( )
- *enable*  
public void enable( boolean )
- *enableEvents*  
protected final void enableEvents( long )
- *enableInputMethods*  
public void enableInputMethods( boolean )



- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent( )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getBackBuffer*  
Image getBackBuffer( )
- *getBackground*  
public Color getBackground( )
- *getBounds*  
public Rectangle getBounds( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy getBufferStrategy( )
- *getColorModel*  
public ColorModel getColorModel( )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener getComponentListeners( )
- *getComponentOrientation*  
public ComponentOrientation getComponentOrientation( )
- *getCursor*  
public Cursor getCursor( )
- *getDropTarget*  
public synchronized DropTarget getDropTarget( )
- *getFocusCycleRootAncestor*  
public Container getFocusCycleRootAncestor( )
- *getFocusListeners*  
public synchronized FocusListener getFocusListeners( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set getFocusTraversalKeys\_NoIDCheck( int )

- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int    )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font    )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class    )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point    )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )

- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )
- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )
- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )

- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )
- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )

- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics )
- *list*  
public void **list**( )
- *list*  
public void **list**( java.io.PrintStream )
- *list*  
public void **list**( java.io.PrintStream , int )
- *list*  
public void **list**( java.io.PrintWriter )
- *list*  
public void **list**( java.io.PrintWriter , int )
- *locate*  
public Component **locate**( int , int )
- *location*  
public Point **location**( )
- *lostFocus*  
public boolean **lostFocus**( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension **minimumSize**( )
- *mouseDown*  
public boolean **mouseDown**( java.awt.Event , int , int )
- *mouseDrag*  
public boolean **mouseDrag**( java.awt.Event , int , int )
- *mouseEnter*  
public boolean **mouseEnter**( java.awt.Event , int , int )
- *mouseExit*  
public boolean **mouseExit**( java.awt.Event , int , int )
- *mouseMove*  
public boolean **mouseMove**( java.awt.Event , int , int )
- *mouseUp*  
public boolean **mouseUp**( java.awt.Event , int , int )
- *move*  
public void **move**( int , int )
- *nextFocus*  
public void **nextFocus**( )
- *nextFocusHelper*  
boolean **nextFocusHelper**( )
- *numListening*  
int **numListening**( long )
- *paint*  
public void **paint**( java.awt.Graphics )
- *paintAll*  
public void **paintAll**( java.awt.Graphics )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics )
- *paramString*  
protected String **paramString**( )
- *postEvent*  
public boolean **postEvent**( java.awt.Event )

- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *prepareImage*  
public boolean **prepareImage**( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean **prepareImage**( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void **print**( java.awt.Graphics )
- *printAll*  
public void **printAll**( java.awt.Graphics )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics )
- *processComponentEvent*  
protected void **processComponentEvent**( java.awt.event.ComponentEvent )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent )
- *processFocusEvent*  
protected void **processFocusEvent**( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void **processHierarchyBoundsEvent**( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void **processHierarchyEvent**( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void **processInputMethodEvent**( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void **processKeyEvent**( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void **processMouseEvent**( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void **processMouseMotionEvent**( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void **processMouseWheelEvent**( java.awt.event.MouseWheelEvent )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *remove*  
public synchronized void **remove**( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void **removeComponentListener**( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void **removeFocusListener**( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void **removeHierarchyBoundsListener**( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void **removeHierarchyListener**( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void **removeInputMethodListener**( java.awt.event.InputMethodListener )

- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *repaint*  
public void repaint( )
- *repaint*  
public void repaint( int , int , int , int )
- *repaint*  
public void repaint( long )
- *repaint*  
public void repaint( long , int , int , int , int )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )

- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )



- *transferFocusUpCycle*  
public void **transferFocusUpCycle**( )
- *update*  
public void **update**( java.awt.Graphics )
- *updateCursorImmediately*  
final void **updateCursorImmediately**( )
- *validate*  
public void **validate**( )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.11 CLASS Main.AboutAction

---

Pop the About dialog

#### DECLARATION

---

```
protected class Main.AboutAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *Main.AboutAction*  
protected **Main.AboutAction**( )

#### METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object **clone**( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object **getKeys**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object **getValue**( java.lang.String )

- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.12 CLASS Main.ClearAction

---

To clear the output

#### DECLARATION

---

```
protected class Main.ClearAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *Main.ClearAction*  
protected Main.ClearAction( )

#### METHODS

---

- *actionPerformed*  
public void actionPerformed( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getKeys*  
public Object getKeys( )

- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object **getValue**( java.lang.String )
- *isEnabled*  
public boolean **isEnabled**( )
- *putValue*  
public void **putValue**( java.lang.String , java.lang.Object )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void **setEnabled**( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.13 CLASS Main.EditorCaretListener

---

To listen to the editor caret

#### DECLARATION

---

```
protected class Main.EditorCaretListener
extends java.lang.Object
implements javax.swing.event.CaretListener
```

#### CONSTRUCTORS

---

- *Main.EditorCaretListener*  
**protected Main.EditorCaretListener**( )

#### METHODS

---

- *caretUpdate*  
**public void caretUpdate**( javax.swing.event.CaretEvent e )

### 13.2.14 CLASS Main.EvalAction

---

To evaluate the content of the buffer

## DECLARATION

---

```
protected class Main.EvalAction
extends javax.swing.AbstractAction
implements koala.dynamicjava.gui.resource.JComponentModifier
```

---

## CONSTRUCTORS

- *Main.EvalAction*  
`protected Main.EvalAction( )`

## METHODS

- *actionPerformed*  
`public void actionPerformed( java.awt.event.ActionEvent ev )`
- *addJComponent*  
`public void addJComponent( javax.swing.JComponent c )`
- *update*  
`protected void update( )`

## METHODS INHERITED FROM CLASS javax.swing.AbstractAction

- 
- *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener(  
java.beans.PropertyChangeListener )`
  - *clone*  
`protected Object clone( )`
  - *firePropertyChange*  
`protected void firePropertyChange( java.lang.String , java.lang.Object ,  
java.lang.Object )`
  - *getKeys*  
`public Object getKeys( )`
  - *getPropertyChangeListeners*  
`public synchronized PropertyChangeListener getPropertyChangeListeners( )`
  - *getValue*  
`public Object getValue( java.lang.String )`
  - *isEnabled*  
`public boolean isEnabled( )`
  - *putValue*  
`public void putValue( java.lang.String , java.lang.Object )`
  - *readObject*  
`private void readObject( java.io.ObjectInputStream )`
  - *removePropertyChangeListener*  
`public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )`
  - *setEnabled*  
`public void setEnabled( boolean )`
  - *writeObject*  
`private void writeObject( java.io.ObjectOutputStream )`

### 13.2.15 CLASS Main.EvalSelectionAction

---

To evaluate the content of the selection

#### DECLARATION

---

```
protected class Main.EvalSelectionAction
extends javax.swing.AbstractAction
implements koala.dynamicjava.gui.resource.JComponentModifier
```

#### CONSTRUCTORS

---

- *Main.EvalSelectionAction*  
 protected **Main.EvalSelectionAction**( )

#### METHODS

---

- *actionPerformed*  
 public void **actionPerformed**( java.awt.event.ActionEvent ev )
- *addJComponent*  
 public void **addJComponent**( javax.swing.JComponent c )
- *update*  
 protected void **update**( )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
 public synchronized void **addPropertyChangeListener**(  
   java.beans.PropertyChangeListener )
- *clone*  
 protected Object **clone**( )
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
   java.lang.Object )
- *getKeys*  
 public Object **getKeys**( )
- *getPropertyChangeListeners*  
 public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
 public Object **getValue**( java.lang.String )
- *isEnabled*  
 public boolean **isEnabled**( )
- *putValue*  
 public void **putValue**( java.lang.String , java.lang.Object )

- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.16 CLASS Main.ExitAction

---

To exit the application

#### DECLARATION

---

```
protected class Main.ExitAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *Main.ExitAction*  
protected Main.ExitAction( )

#### METHODS

---

- *actionPerformed*  
public void actionPerformed( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getValue*  
public Object getValue( java.lang.String )

- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.17 CLASS Main.InterpreterThread

---

To run the interpreter

#### DECLARATION

---

```
protected class Main.InterpreterThread
extends java.lang.Thread
```

#### CONSTRUCTORS

---

- *Main.InterpreterThread*  
**Main.InterpreterThread( java.io.Reader r )**

#### METHODS

---

- *run*  
**public void run( )**

#### METHODS INHERITED FROM CLASS java.lang.Thread

---

- *<clinit>*  
static void <clinit>( )
- *activeCount*  
public static int activeCount( )
- *blockedOn*  
private void blockedOn( sun.nio.ch.Interruptible )
- *checkAccess*  
public final void checkAccess( )
- *countStackFrames*  
public native int countStackFrames( )

- *currentThread*  
public static native Thread currentThread( )
- *destroy*  
public void destroy( )
- *dumpStack*  
public static void dumpStack( )
- *enumerate*  
public static int enumerate( java.lang.Thread [] )
- *exit*  
private void exit( )
- *getContextClassLoader*  
public ClassLoader getContextClassLoader( )
- *getName*  
public final String getName( )
- *getPriority*  
public final int getPriority( )
- *getThreadGroup*  
public final ThreadGroup getThreadGroup( )
- *holdsLock*  
public static native boolean holdsLock( java.lang.Object )
- *init*  
private void init( java.lang.ThreadGroup , java.lang.Runnable , java.lang.String , long )
- *interrupt*  
public void interrupt( )
- *interrupt0*  
private native void interrupt0( )
- *interrupted*  
public static boolean interrupted( )
- *isAlive*  
public final native boolean isAlive( )
- *isDaemon*  
public final boolean isDaemon( )
- *isInterrupted*  
public boolean isInterrupted( )
- *isInterrupted*  
private native boolean isInterrupted( boolean )
- *join*  
public final void join( )
- *join*  
public final synchronized void join( long )
- *join*  
public final synchronized void join( long , int )
- *nextThreadNum*  
private static synchronized int nextThreadNum( )
- *registerNatives*  
private static native void registerNatives( )
- *resume*  
public final void resume( )
- *resume0*  
private native void resume0( )



- *run*  
public void run( )
- *setContextClassLoader*  
public void setContextClassLoader( java.lang.ClassLoader )
- *setDaemon*  
public final void setDaemon( boolean )
- *setName*  
public final void setName( java.lang.String )
- *setPriority*  
public final void setPriority( int )
- *setPriority0*  
private native void setPriority0( int )
- *sleep*  
public static native void sleep( long )
- *sleep*  
public static void sleep( long , int )
- *start*  
public synchronized native void start( )
- *stop*  
public final void stop( )
- *stop*  
public final synchronized void stop( java.lang.Throwable )
- *stop0*  
private native void stop0( java.lang.Object )
- *suspend*  
public final void suspend( )
- *suspend0*  
private native void suspend0( )
- *toString*  
public String toString( )
- *yield*  
public static native void yield( )

### 13.2.18 CLASS Main.OptionsAction

---

To pop the Options dialog

#### DECLARATION

---

```
protected class Main.OptionsAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *Main.OptionsAction*  
protected Main.OptionsAction( )

## METHODS

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

## METHODS INHERITED FROM CLASS javax.swing.AbstractAction

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

## 13.2.19 CLASS Main.ReinitAction

Reinitializes the interpreter

## DECLARATION

```
protected class Main.ReinitAction
extends javax.swing.AbstractAction
```

## CONSTRUCTORS

- *Main.ReinitAction*  
protected **Main.ReinitAction**( )

## METHODS

- *actionPerformed*  
`public void actionPerformed( java.awt.event.ActionEvent e )`

## METHODS INHERITED FROM CLASS javax.swing.AbstractAction

- *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener(  
java.beans.PropertyChangeListener )`
- *clone*  
`protected Object clone( )`
- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String , java.lang.Object ,  
java.lang.Object )`
- *getKeys*  
`public Object getKeys( )`
- *getPropertyChangeListeners*  
`public synchronized PropertyChangeListener getPropertyChangeListeners( )`
- *getValue*  
`public Object getValue( java.lang.String )`
- *isEnabled*  
`public boolean isEnabled( )`
- *putValue*  
`public void putValue( java.lang.String , java.lang.Object )`
- *readObject*  
`private void readObject( java.io.ObjectInputStream )`
- *removePropertyChangeListener*  
`public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )`
- *setEnabled*  
`public void setEnabled( boolean )`
- *writeObject*  
`private void writeObject( java.io.ObjectOutputStream )`

## 13.2.20 CLASS Main.ScrollBarModelChangeListener

To listen to the changes in the output area vertical scroll bar model

## DECLARATION

```
protected class Main.ScrollBarModelChangeListener
extends java.lang.Object
implements javax.swing.event.ChangeListener
```

CONSTRUCTORS

---

- *Main.ScrollBarModelChangeListener*  
protected **Main.ScrollBarModelChangeListener**( )

METHODS

---

- *stateChanged*  
public void **stateChanged**( javax.swing.event.ChangeEvent e )

**13.2.21 CLASS Main.StopAction**

---

To stop the interpreter thread

DECLARATION

---

```
protected class Main.StopAction
extends javax.swing.AbstractAction
implements koala.dynamicjava.gui.resource.JComponentModifier
```

CONSTRUCTORS

---

- *Main.StopAction*  
protected **Main.StopAction**( )

METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent ev )
- *addJComponent*  
public void **addJComponent**( javax.swing.JComponent c )
- *update*  
protected void **update**( )

METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object **clone**( )

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object **getKeys**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object **getValue**( java.lang.String )
- *isEnabled*  
public boolean **isEnabled**( )
- *putValue*  
public void **putValue**( java.lang.String , java.lang.Object )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void **setEnabled**( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.22 CLASS OptionsDialog

---

The 'options' dialog

#### DECLARATION

---

```
public class OptionsDialog
extends javax.swing.JDialog
implements koala.dynamicjava.gui.resource.ActionMap
```

#### CONSTRUCTORS

---

- *OptionsDialog*  
public **OptionsDialog**( koala.dynamicjava.gui.Main owner )
  - **Usage**
    - \* Creates a new dialog
  - **Parameters**
    - \* **owner** - the owner of this dialog

METHODS

---

- *createButtonsPanel*  
`protected JPanel createButtonsPanel( )`
  - **Usage**
    - \* Creates the buttons panel

---
- *createClassPathPanel*  
`protected JPanel createClassPathPanel( )`
  - **Usage**
    - \* Creates the classpath panel

---
- *createGeneralPanel*  
`protected JPanel createGeneralPanel( )`
  - **Usage**
    - \* Creates the general panel

---
- *createLibraryPathPanel*  
`protected JPanel createLibraryPathPanel( )`
  - **Usage**
    - \* Creates the library path panel

---
- *createPathPanel*  
`protected JPanel createPathPanel( )`
  - **Usage**
    - \* Creates the path panel

---
- *createTabbedPane*  
`protected JTabbedPane createTabbedPane( )`
  - **Usage**
    - \* Creates the tabbed pane

---
- *getAction*  
`public Action getAction( java.lang.String key )`
  - **Usage**
    - \* Returns the action associated with the given string or null on error
  - **Parameters**
    - \* **key** - the key mapped with the action to get
  - **Exceptions**
    - \* `koala.dynamicjava.gui.resource.MissingListenerException` - if the action is not found

---
- *getClassPath*  
`public String getClassPath( )`

- **Usage**
  - \* Returns the classpaths

---
- *getGUIName*
  - public String **getGUIName**( )
  - **Usage**
  - \* Returns the GUI name

---
- *getInitializationFilename*
  - public String **getInitializationFilename**( )
  - **Usage**
  - \* The initialization file name

---
- *getInterpreterName*
  - public String **getInterpreterName**( )
  - **Usage**
  - \* Returns the interpreter name

---
- *getLibraryPath*
  - public String **getLibraryPath**( )
  - **Usage**
  - \* Returns the library paths

---
- *getOptions*
  - public OptionsDialog.OptionSet **getOptions**( )
  - **Usage**
  - \* Returns an object that holds the current options

---
- *getStartupInitializationFilename*
  - public String **getStartupInitializationFilename**( )
  - **Usage**
  - \* The startup initialization file name

---
- *isErrorSelected*
  - public boolean **isErrorSelected**( )
  - **Usage**
  - \* Has the standard error to be redirected?

---
- *isGUIDefined*
  - public boolean **isGUIDefined**( )
  - **Usage**
  - \* Has the GUI to be defined?

---
- *isInitializationSelected*
  - public boolean **isInitializationSelected**( )

- **Usage**
  - \* Is the initialization file option selected?

---
- *isInterpreterDefined*  
**public boolean isInterpreterDefined( )**
  - **Usage**
    - \* Has the interpreter to be defined?

---
- *isOutputSelected*  
**public boolean isOutputSelected( )**
  - **Usage**
    - \* Has the output to be redirected?

---
- *isStartupInitializationSelected*  
**public boolean isStartupInitializationSelected( )**
  - **Usage**
    - \* Is the startup initialization file option selected?

---
- *setConstraintsCoords*  
**protected static void setConstraintsCoords( java.awt.GridBagConstraints constraints, int x, int y, int width, int height )**


---
- *setOptions*  
**public void setOptions( koala.dynamicjava.gui.OptionsDialog.OptionSet optionSet )**
  - **Usage**
    - \* Sets the options according to the given option set

#### METHODS INHERITED FROM CLASS javax.swing.JDialog

---

- *<clinit>*  
**static void <clinit>( )**


---
- *addImpl*  
**protected void addImpl( java.awt.Component , java.lang.Object , int )**


---
- *createRootPane*  
**protected JRootPane createRootPane( )**


---
- *createRootPaneException*  
**private Error createRootPaneException( java.lang.String )**


---
- *dialogInit*  
**protected void dialogInit( )**


---
- *getAccessibleContext*  
**public AccessibleContext getAccessibleContext( )**


---
- *getContentPane*  
**public Container getContentPane( )**


---
- *getDefaultCloseOperation*  
**public int getDefaultCloseOperation( )**


---



- *getGlassPane*  
public Component **getGlassPane**( )
- *getJMenuBar*  
public JMenuBar **getJMenuBar**( )
- *getLayeredPane*  
public JLayeredPane **getLayeredPane**( )
- *getRootPane*  
public JRootPane **getRootPane**( )
- *isDefaultLookAndFeelDecorated*  
public static boolean **isDefaultLookAndFeelDecorated**( )
- *isRootPaneCheckingEnabled*  
protected boolean **isRootPaneCheckingEnabled**( )
- *paramString*  
protected String **paramString**( )
- *processWindowEvent*  
protected void **processWindowEvent**( java.awt.event.WindowEvent    )
- *remove*  
public void **remove**( java.awt.Component    )
- *setContentPane*  
public void **setContentPane**( java.awt.Container    )
- *setDefaultCloseOperation*  
public void **setDefaultCloseOperation**( int    )
- *setDefaultLookAndFeelDecorated*  
public static void **setDefaultLookAndFeelDecorated**( boolean    )
- *setGlassPane*  
public void **setGlassPane**( java.awt.Component    )
- *setJMenuBar*  
public void **setJMenuBar**( javax.swing.JMenuBar    )
- *setLayeredPane*  
public void **setLayeredPane**( javax.swing.JLayeredPane    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setRootPane*  
protected void **setRootPane**( javax.swing.JRootPane    )
- *setRootPaneCheckingEnabled*  
protected void **setRootPaneCheckingEnabled**( boolean    )
- *update*  
public void **update**( java.awt.Graphics    )

#### METHODS INHERITED FROM CLASS java.awt.Dialog

---

- *<clinit>*  
static void **<clinit>**( )
- *addNotify*  
public void **addNotify**( )
- *conditionalShow*  
private boolean **conditionalShow**( )
- *constructComponentName*  
String **constructComponentName**( )

- *dispose*  
public void dispose( )
- *disposeImpl*  
private void disposeImpl( )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getTitle*  
public String getTitle( )
- *hide*  
public void hide( )
- *hideAndDisposeHandler*  
private void hideAndDisposeHandler( )
- *initIDs*  
private static native void initIDs( )
- *interruptBlocking*  
void interruptBlocking( )
- *isModal*  
public boolean isModal( )
- *isResizable*  
public boolean isResizable( )
- *isUndecorated*  
public boolean isUndecorated( )
- *paramString*  
protected String paramString( )
- *setModal*  
public void setModal( boolean )
- *setResizable*  
public void setResizable( boolean )
- *setTitle*  
public void setTitle( java.lang.String )
- *setUndecorated*  
public void setUndecorated( boolean )
- *show*  
public void show( )

#### METHODS INHERITED FROM CLASS java.awt.Window

---

- *<clinit>*  
static void <clinit>( )
- *addNotify*  
public void addNotify( )
- *addOwnedWindow*  
void addOwnedWindow( java.lang.ref.WeakReference )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *addWindowFocusListener*  
public synchronized void addWindowFocusListener( java.awt.event.WindowFocusListener )

- *addWindowListener*  
public synchronized void addWindowListener( java.awt.event.WindowListener )
- *addWindowStateListener*  
public synchronized void addWindowStateListener( java.awt.event.WindowStateListener )
- *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyResourceBundle*  
public void applyResourceBundle( java.util.ResourceBundle )
- *applyResourceBundle*  
public void applyResourceBundle( java.lang.String )
- *clearMostRecentFocusOwnerOnHide*  
final void clearMostRecentFocusOwnerOnHide( )
- *connectOwnedWindow*  
void connectOwnedWindow( java.awt.Window )
- *constructComponentName*  
String constructComponentName( )
- *createBufferStrategy*  
public void createBufferStrategy( int )
- *createBufferStrategy*  
public void createBufferStrategy( int , java.awt.BufferCapabilities )
- *deliverMouseWheelToAncestor*  
void deliverMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *dispose*  
public void dispose( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *finalize*  
protected void finalize( )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getBufferStrategy*  
public BufferStrategy getBufferStrategy( )
- *getFocusableWindowState*  
public boolean getFocusableWindowState( )
- *getFocusCycleRootAncestor*  
public final Container getFocusCycleRootAncestor( )
- *getFocusOwner*  
public Component getFocusOwner( )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getInputContext*  
public InputContext getInputContext( )

- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocale*  
public Locale **getLocale**( )
- *getMostRecentFocusOwner*  
public Component **getMostRecentFocusOwner**( )
- *getOwnedWindows*  
public Window **getOwnedWindows**( )
- *getOwner*  
public Window **getOwner**( )
- *getTemporaryLostComponent*  
Component **getTemporaryLostComponent**( )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getWarningString*  
public final String **getWarningString**( )
- *getWindowFocusListeners*  
public synchronized WindowFocusListener **getWindowFocusListeners**( )
- *getWindowListeners*  
public synchronized WindowListener **getWindowListeners**( )
- *getWindowStateListeners*  
public synchronized WindowStateListener **getWindowStateListeners**( )
- *hide*  
public void **hide**( )
- *init*  
private void **init**( java.awt.GraphicsConfiguration )
- *initIDs*  
private static native void **initIDs**( )
- *isActive*  
public boolean **isActive**( )
- *isFocusableWindow*  
public final boolean **isFocusableWindow**( )
- *isFocusCycleRoot*  
public final boolean **isFocusCycleRoot**( )
- *isFocused*  
public boolean **isFocused**( )
- *isShowing*  
public boolean **isShowing**( )
- *ownedInit*  
private void **ownedInit**( java.awt.Window )
- *pack*  
public void **pack**( )
- *postEvent*  
public boolean **postEvent**( java.awt.Event )
- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent )
- *postWindowEvent*  
synchronized void **postWindowEvent**( int )
- *preProcessKeyEvent*  
void **preProcessKeyEvent**( java.awt.event.KeyEvent )

- *processEvent*  
protected void processEvent( java.awt.AWTEvent    )
- *processWindowEvent*  
protected void processWindowEvent( java.awt.event.WindowEvent    )
- *processWindowFocusEvent*  
protected void processWindowFocusEvent( java.awt.event.WindowEvent    )
- *processWindowStateEvent*  
protected void processWindowStateEvent( java.awt.event.WindowEvent    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *removeOwnedWindow*  
void removeOwnedWindow( java.lang.ref.WeakReference    )
- *removeWindowFocusListener*  
public synchronized void removeWindowFocusListener( java.awt.event.WindowFocusListener    )
- *removeWindowListener*  
public synchronized void removeWindowListener( java.awt.event.WindowListener    )
- *removeWindowStateListener*  
public synchronized void removeWindowStateListener( java.awt.event.WindowStateListener    )
- *resetGC*  
void resetGC(    )
- *setCursor*  
public void setCursor( java.awt.Cursor    )
- *setFocusableWindowState*  
public void setFocusableWindowState( boolean    )
- *setFocusCycleRoot*  
public final void setFocusCycleRoot( boolean    )
- *setLocationRelativeTo*  
public void setLocationRelativeTo( java.awt.Component    )
- *setTemporaryLostComponent*  
Component setTemporaryLostComponent( java.awt.Component    )
- *setWarningString*  
private void setWarningString(    )
- *show*  
public void show(    )
- *toBack*  
public void toBack(    )
- *toFront*  
public void toFront(    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

METHODS INHERITED FROM CLASS `java.awt.Container`

- *<clinit>*  
`static void <clinit>( )`
- *add*  
`public Component add( java.awt.Component )`
- *add*  
`public Component add( java.awt.Component , int )`
- *add*  
`public void add( java.awt.Component , java.lang.Object )`
- *add*  
`public void add( java.awt.Component , java.lang.Object , int )`
- *add*  
`public Component add( java.lang.String , java.awt.Component )`
- *addContainerListener*  
`public synchronized void addContainerListener( java.awt.event.ContainerListener )`
- *addImpl*  
`protected void addImpl( java.awt.Component , java.lang.Object , int )`
- *addNotify*  
`public void addNotify( )`
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener )`
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
- *adjustDecendantsOnParent*  
`void adjustDecendantsOnParent( int )`
- *adjustDescendants*  
`void adjustDescendants( int )`
- *adjustListeningChildren*  
`void adjustListeningChildren( long , int )`
- *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
- *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
- *checkGD*  
`void checkGD( java.lang.String )`
- *clearCurrentFocusCycleRootOnHide*  
`void clearCurrentFocusCycleRootOnHide( )`
- *clearMostRecentFocusOwnerOnHide*  
`void clearMostRecentFocusOwnerOnHide( )`
- *containsFocus*  
`final boolean containsFocus( )`
- *countComponents*  
`public int countComponents( )`
- *countHierarchyMembers*  
`int countHierarchyMembers( )`
- *createChildHierarchyEvents*  
`void createChildHierarchyEvents( int , long , boolean )`

- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )

- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseEventTarget*  
Component getMouseEventTarget( int , int , boolean )
- *getMouseEventTarget*  
private Component getMouseEventTarget( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component getMouseEventTargetImpl( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *insets*  
public Insets insets( )
- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )



- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )
- *printComponents*  
public void printComponents( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processContainerEvent*  
protected void processContainerEvent( java.awt.event.ContainerEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *proxyEnableEvents*  
void proxyEnableEvents( long )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public void remove( java.awt.Component )
- *remove*  
public void remove( int )
- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )

- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy )
- *setFont*  
public void **setFont**( java.awt.Font )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager )
- *setZOrder*  
void **setZOrder**( java.awt.Component , int )
- *transferFocusBackward*  
public void **transferFocusBackward**( )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**( )
- *update*  
public void **update**( java.awt.Graphics )
- *validate*  
public void **validate**( )
- *validateTree*  
protected void **validateTree**( )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void **<clinit>**( )
- *action*  
public boolean **action**( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void **add**( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void **addComponentListener**( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void **addFocusListener**( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void **addHierarchyBoundsListener**( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void **addHierarchyListener**( java.awt.event.HierarchyListener )
- *addInputMethodListener*  
public synchronized void **addInputMethodListener**( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void **addKeyListener**( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void **addMouseListener**( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void **addMouseMotionListener**( java.awt.event.MouseMotionListener )

- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *areInputMethodsEnabled*  
boolean areInputMethodsEnabled( )
- *autoProcessMouseWheel*  
void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )
- *autoTransferFocus*  
final void autoTransferFocus( boolean )
- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )

- *createBufferStrategy*  
void **createBufferStrategy**( int    )
- *createBufferStrategy*  
void **createBufferStrategy**( int    , java.awt.BufferCapabilities    )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int    , long    , boolean    )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int    , java.awt.Component    , java.awt.Container    , long    , boolean    )
- *createImage*  
public Image **createImage**( java.awt.image.ImageProducer    )
- *createImage*  
public Image **createImage**( int    , int    )
- *createVolatileImage*  
public VolatileImage **createVolatileImage**( int    , int    )
- *createVolatileImage*  
public VolatileImage **createVolatileImage**( int    , int    , java.awt.ImageCapabilities    )
- *deliverEvent*  
public void **deliverEvent**( java.awt.Event    )
- *disable*  
public void **disable**( )
- *disableEvents*  
protected final void **disableEvents**( long    )
- *dispatchEvent*  
public final void **dispatchEvent**( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean **dispatchMouseWheelToAncestor**( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void **doAutoTransfer**( boolean    )
- *doLayout*  
public void **doLayout**( )
- *enable*  
public void **enable**( )
- *enable*  
public void **enable**( boolean    )
- *enableEvents*  
protected final void **enableEvents**( long    )
- *enableInputMethods*  
public void **enableInputMethods**( boolean    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean **eventTypeEnabled**( int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , int    , int    )

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getAccessibleIndexInParent*  
int **getAccessibleIndexInParent**( )
- *getAccessibleStateSet*  
AccessibleStateSet **getAccessibleStateSet**( )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )
- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )

- *getFontMetrics*  
public FontMetrics getFontMetrics( java.awt.Font    )
- *getForeground*  
public Color getForeground( )
- *getGraphics*  
public Graphics getGraphics( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getHeight*  
public int getHeight( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener getHierarchyBoundsListeners( )
- *getHierarchyListeners*  
public synchronized HierarchyListener getHierarchyListeners( )
- *getIgnoreRepaint*  
public boolean getIgnoreRepaint( )
- *getInputContext*  
public InputContext getInputContext( )
- *getInputMethodListeners*  
public synchronized InputMethodListener getInputMethodListeners( )
- *getInputMethodRequests*  
public InputMethodRequests getInputMethodRequests( )
- *getKeyListeners*  
public synchronized KeyListener getKeyListeners( )
- *getListeners*  
public EventListener getListeners( java.lang.Class    )
- *getLocale*  
public Locale getLocale( )
- *getLocation*  
public Point getLocation( )
- *getLocation*  
public Point getLocation( java.awt.Point    )
- *getLocationOnScreen\_NoTreeLock*  
final Point getLocationOnScreen\_NoTreeLock( )
- *getLocationOnScreen*  
public Point getLocationOnScreen( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseListeners*  
public synchronized MouseListener getMouseListeners( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )

- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )
- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )

- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )



- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )
- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )

- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )

- *removeMouseListener*  
public synchronized void removeMouseListener(  
java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
protected boolean requestFocus( boolean    )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean    , boolean    )
- *requestFocusInWindow*  
public boolean requestFocusInWindow(    )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean    )
- *resetGC*  
void resetGC(    )
- *reshape*  
public void reshape( int    , int    , int    , int    )
- *resize*  
public void resize( java.awt.Dimension    )
- *resize*  
public void resize( int    , int    )
- *setBackground*  
public void setBackground( java.awt.Color    )
- *setBounds*  
public void setBounds( int    , int    , int    , int    )
- *setBounds*  
public void setBounds( java.awt.Rectangle    )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation    )
- *setCursor*  
public void setCursor( java.awt.Cursor    )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget    )
- *setEnabled*  
public void setEnabled( boolean    )

- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusUpCycle*  
public void transferFocusUpCycle( )
- *update*  
public void update( java.awt.Graphics )
- *updateCursorImmediately*  
final void updateCursorImmediately( )
- *validate*  
public void validate( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.23 CLASS OptionsDialog.CancelButtonAction

---

The action associated with the 'Cancel' button

#### DECLARATION

---

```
protected class OptionsDialog.CancelButtonAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *OptionsDialog.CancelButtonAction*  
protected **OptionsDialog.CancelButtonAction**( )

#### METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.24 CLASS OptionsDialog.CPLAddButtonAction

---

The action associated with the 'add' button of the class path panel

#### DECLARATION

---

```
protected class OptionsDialog.CPLAddButtonAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *OptionsDialog.CPLAddButtonAction*  
protected **OptionsDialog.CPLAddButtonAction**( )

#### METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.25 CLASS *OptionsDialog.GUIPanel*

---

The GUI option panel

#### DECLARATION

---

```
protected class OptionsDialog.GUIPanel
extends javax.swing.JPanel
```

#### CONSTRUCTORS

---

- *OptionsDialog.GUIPanel*  
**public OptionsDialog.GUIPanel( )**
  - **Usage**
    - \* Creates a new panel

#### METHODS

---

- *getFilename*  
**public String getFilename( )**
  - **Usage**
    - \* Returns the initialization file name
- *getName*  
**public String getName( )**
  - **Usage**
    - \* Returns the name to give to the GUI
- *isErrorSelected*  
**public boolean isErrorSelected( )**
  - **Usage**
    - \* Has the error to be redirected?
- *isInitializationSelected*  
**public boolean isInitializationSelected( )**
  - **Usage**
    - \* Is the initialization file checkbox selected
- *isOutputSelected*  
**public boolean isOutputSelected( )**
  - **Usage**
    - \* Has the output to be redirected?

- 
- *isSelected*  
 public boolean **isSelected**( )  
 – **Usage**  
 \* Has the GUI to be exported?

---

  - *setErrorSelected*  
 public void **setErrorSelected**( boolean b )  
 – **Usage**  
 \* Sets the state of the error checkbox

---

  - *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
 \* Sets the initialization file name

---

  - *setInitializationSelected*  
 public void **setInitializationSelected**( boolean b )  
 – **Usage**  
 \* Sets the initialization file checkbox state

---

  - *setName*  
 public void **setName**( java.lang.String s )  
 – **Usage**  
 \* Sets the GUI name

---

  - *setOutputSelected*  
 public void **setOutputSelected**( boolean b )  
 – **Usage**  
 \* Sets the state of the output checkbox

---

  - *setSelected*  
 public void **setSelected**( boolean b )  
 – **Usage**  
 \* Sets the state of the checkbox

#### METHODS INHERITED FROM CLASS javax.swing.JPanel

---

- *getAccessibleContext*  
 public AccessibleContext **getAccessibleContext**( )

---

- *getUI*  
 public PanelUI **getUI**( )

---

- *getUIClassID*  
 public String **getUIClassID**( )

---



- *paramString*  
protected String paramString( )
- *setUI*  
public void setUI( javax.swing.plaf.PanelUI )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

---

#### METHODS INHERITED FROM CLASS javax.swing.JComponent

---

- *\_paintImmediately*  
void \_paintImmediately( int , int , int , int )
- *<clinit>*  
static void <clinit>( )
- *addAncestorListener*  
public void addAncestorListener( javax.swing.event.AncestorListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *addVetoableChangeListener*  
public synchronized void addVetoableChangeListener( java.beans.VetoableChangeListener )
- *adjustPaintFlags*  
private void adjustPaintFlags( )
- *alwaysOnTop*  
boolean alwaysOnTop( )
- *checkIfChildObscuredBySibling*  
boolean checkIfChildObscuredBySibling( )
- *componentInputMapChanged*  
void componentInputMapChanged( javax.swing.ComponentInputMap )
- *computeVisibleRect*  
static final void computeVisibleRect( java.awt.Component , java.awt.Rectangle )
- *computeVisibleRect*  
public void computeVisibleRect( java.awt.Rectangle )
- *compWriteObjectNotify*  
void compWriteObjectNotify( )
- *contains*  
public boolean contains( int , int )
- *createToolTip*  
public JToolTip createToolTip( )
- *deregisterNextFocusableComponent*  
private void deregisterNextFocusableComponent( )
- *disable*  
public void disable( )

- *enable*  
public void enable( )
- *enableSerialization*  
void enableSerialization( )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , byte , byte )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , char , char )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getAncestorListeners*  
public AncestorListener getAncestorListeners( )
- *getAutoscrolls*  
public boolean getAutoscrolls( )
- *getBorder*  
public Border getBorder( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary getClientProperties( )
- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )

- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )
- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )

- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )
- *getToolTipText*  
public String getToolTipText( )
- *getToolTipText*  
public String getToolTipText( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container getTopLevelAncestor( )
- *getTransferHandler*  
public TransferHandler getTransferHandler( )
- *getUIClassID*  
public String getUIClassID( )
- *getVerifyInputWhenFocusTarget*  
public boolean getVerifyInputWhenFocusTarget( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener getVetoableChangeListeners( )
- *getWidth*  
public int getWidth( )
- *getVisibleRect*  
public Rectangle getVisibleRect( )
- *getWriteObjCounter*  
static byte getWriteObjCounter( javax.swing.JComponent )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *grabFocus*  
public void grabFocus( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component )
- *isManagingFocus*  
public boolean isManagingFocus( )
- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )

- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )
- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )

- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent    )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent    )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object    , java.lang.Object    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int    , int    , int    , int    )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener    ,  
javax.swing.KeyStroke    , int    )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener    ,  
java.lang.String    , javax.swing.KeyStroke    , int    )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent(    )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component    )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean    )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke    )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener(  
java.beans.VetoableChangeListener    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *repaint*  
public void repaint( java.awt.Rectangle    )
- *requestDefaultFocus*  
public boolean requestDefaultFocus(    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
public boolean requestFocus( boolean    )
- *requestFocusInWindow*  
public boolean requestFocusInWindow(    )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean    )

- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )
- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )

- *setPaintingChild*  
void **setPaintingChild**( java.awt.Component    )
- *setPreferredSize*  
public void **setPreferredSize**( java.awt.Dimension    )
- *setRequestFocusEnabled*  
public void **setRequestFocusEnabled**( boolean    )
- *setToolTipText*  
public void **setToolTipText**( java.lang.String    )
- *setTransferHandler*  
public void **setTransferHandler**( javax.swing.TransferHandler    )
- *setUI*  
protected void **setUI**( javax.swing.plaf.ComponentUI    )
- *setVerifyInputWhenFocusTarget*  
public void **setVerifyInputWhenFocusTarget**( boolean    )
- *setVisible*  
public void **setVisible**( boolean    )
- *setWriteObjCounter*  
static void **setWriteObjCounter**( javax.swing.JComponent    , byte    )
- *shouldDebugGraphics*  
int **shouldDebugGraphics**(    )
- *superProcessMouseEvent*  
void **superProcessMouseEvent**( java.awt.event.MouseEvent    )
- *unregisterKeyboardAction*  
public void **unregisterKeyboardAction**( javax.swing.KeyStroke    )
- *unregisterWithKeyboardManager*  
private void **unregisterWithKeyboardManager**(    )
- *unregisterWithKeyboardManager*  
private void **unregisterWithKeyboardManager**( javax.swing.KeyStroke    )
- *update*  
public void **update**( java.awt.Graphics    )
- *updateUI*  
public void **updateUI**(    )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void **<clinit>**(    )
- *add*  
public Component **add**( java.awt.Component    )
- *add*  
public Component **add**( java.awt.Component    , int    )
- *add*  
public void **add**( java.awt.Component    , java.lang.Object    )
- *add*  
public void **add**( java.awt.Component    , java.lang.Object    , int    )
- *add*  
public Component **add**( java.lang.String    , java.awt.Component    )



- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int )
- *adjustDescendants*  
void adjustDescendants( int )
- *adjustListeningChildren*  
void adjustListeningChildren( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *checkGD*  
void checkGD( java.lang.String )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *containsFocus*  
final boolean containsFocus( )
- *countComponents*  
public int countComponents( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )

- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )
- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseEventTarget*  
Component getMouseEventTarget( int , int , boolean )
- *getMouseEventTarget*  
private Component getMouseEventTarget( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )

- *getMouseEventTargetImpl*  
private Component getMouseEventTargetImpl( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *insets*  
public Insets insets( )
- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )

- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent    )
- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *preProcessKeyEvent*  
void **preProcessKeyEvent**( java.awt.event.KeyEvent    )
- *print*  
public void **print**( java.awt.Graphics    )
- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )
- *removeAll*  
public void **removeAll**( )
- *removeContainerListener*  
public synchronized void **removeContainerListener**( java.awt.event.ContainerListener  
  )
- *removeNotify*  
public void **removeNotify**( )
- *setFocusCycleRoot*  
public void **setFocusCycleRoot**( boolean    )
- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int    , java.util.Set    )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy    )
- *setFont*  
public void **setFont**( java.awt.Font    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setZOrder*  
void **setZOrder**( java.awt.Component    , int    )
- *transferFocusBackward*  
public void **transferFocusBackward**( )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**( )
- *update*  
public void **update**( java.awt.Graphics    )

- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

---

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )
- *addInputMethodListener*  
public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )
- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )

- *areInputMethodsEnabled*  
boolean **areInputMethodsEnabled**( )
- *autoProcessMouseWheel*  
void **autoProcessMouseWheel**( java.awt.event.MouseWheelEvent )
- *autoTransferFocus*  
final void **autoTransferFocus**( boolean )
- *bounds*  
public Rectangle **bounds**( )
- *checkGD*  
void **checkGD**( java.lang.String )
- *checkImage*  
public int **checkImage**( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int **checkImage**( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean **checkWindowClosingException**( )
- *clearCurrentFocusCycleRootOnHide*  
void **clearCurrentFocusCycleRootOnHide**( )
- *clearMostRecentFocusOwnerOnHide*  
void **clearMostRecentFocusOwnerOnHide**( )
- *coalesceEvents*  
protected AWTEvent **coalesceEvents**( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String **constructComponentName**( )
- *contains*  
public boolean **contains**( int , int )
- *contains*  
public boolean **contains**( java.awt.Point )
- *containsFocus*  
boolean **containsFocus**( )
- *countHierarchyMembers*  
int **countHierarchyMembers**( )
- *createBufferStrategy*  
void **createBufferStrategy**( int )
- *createBufferStrategy*  
void **createBufferStrategy**( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int , long , boolean )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image **createImage**( java.awt.image.ImageProducer )
- *createImage*  
public Image **createImage**( int , int )
- *createVolatileImage*  
public VolatileImage **createVolatileImage**( int , int )
- *createVolatileImage*  
public VolatileImage **createVolatileImage**( int , int , java.awt.ImageCapabilities )

- *deliverEvent*  
public void deliverEvent( java.awt.Event    )
- *disable*  
public void disable(    )
- *disableEvents*  
protected final void disableEvents( long    )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void doAutoTransfer( boolean    )
- *doLayout*  
public void doLayout(    )
- *enable*  
public void enable(    )
- *enable*  
public void enable( boolean    )
- *enableEvents*  
protected final void enableEvents( long    )
- *enableInputMethods*  
public void enableInputMethods( boolean    )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int    )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String    , int    , int    )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String    , java.lang.Object    , java.lang.Object    )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext(    )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent(    )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet(    )
- *getAlignmentX*  
public float getAlignmentX(    )
- *getAlignmentY*  
public float getAlignmentY(    )
- *getBackBuffer*  
Image getBackBuffer(    )
- *getBackground*  
public Color getBackground(    )
- *getBounds*  
public Rectangle getBounds(    )

- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy getBufferStrategy( )
- *getColorModel*  
public ColorModel getColorModel( )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener getComponentListeners( )
- *getComponentOrientation*  
public ComponentOrientation getComponentOrientation( )
- *getCursor*  
public Cursor getCursor( )
- *getDropTarget*  
public synchronized DropTarget getDropTarget( )
- *getFocusCycleRootAncestor*  
public Container getFocusCycleRootAncestor( )
- *getFocusListeners*  
public synchronized FocusListener getFocusListeners( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set getFocusTraversalKeys\_NoIDCheck( int )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalKeysEnabled*  
public boolean getFocusTraversalKeysEnabled( )
- *getFont\_NoClientCode*  
final Font getFont\_NoClientCode( )
- *getFont*  
public Font getFont( )
- *getFontMetrics*  
public FontMetrics getFontMetrics( java.awt.Font )
- *getForeground*  
public Color getForeground( )
- *getGraphics*  
public Graphics getGraphics( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getHeight*  
public int getHeight( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener getHierarchyBoundsListeners( )
- *getHierarchyListeners*  
public synchronized HierarchyListener getHierarchyListeners( )
- *getIgnoreRepaint*  
public boolean getIgnoreRepaint( )
- *getInputContext*  
public InputContext getInputContext( )



- *getInputMethodListeners*  
public synchronized InputMethodListener getInputMethodListeners( )
- *getInputMethodRequests*  
public InputMethodRequests getInputMethodRequests( )
- *getKeyListeners*  
public synchronized KeyListener getKeyListeners( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocale*  
public Locale getLocale( )
- *getLocation*  
public Point getLocation( )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point getLocationOnScreen\_NoTreeLock( )
- *getLocationOnScreen*  
public Point getLocationOnScreen( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseListeners*  
public synchronized MouseListener getMouseListeners( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )
- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )

- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )
- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )

- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )

- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )
- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )

- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent    )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent    )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent    )
- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *remove*  
public synchronized void remove( java.awt.MenuComponent    )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener    )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener    )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener    )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener    )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener    )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener    )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener    )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener    )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    , java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )

- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )

- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusUpCycle*  
public void transferFocusUpCycle( )
- *update*  
public void update( java.awt.Graphics )
- *updateCursorImmediately*  
final void updateCursorImmediately( )
- *validate*  
public void validate( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.26 CLASS OptionsDialog.GUIPanel.CheckBoxChangeListener

---

To listen to the checkbox

#### DECLARATION

---

```
protected class OptionsDialog.GUIPanel.CheckBoxChangeListener
extends java.lang.Object
implements javax.swing.event.ChangeListener
```

#### CONSTRUCTORS

---

- *OptionsDialog.GUIPanel.CheckBoxChangeListener*  
protected **OptionsDialog.GUIPanel.CheckBoxChangeListener( )**

METHODS

---

- *stateChanged*  
public void **stateChanged**( javax.swing.event.ChangeEvent e )

**13.2.27 CLASS OptionsDialog.GUIPanel.FileCheckBoxChangeListener**

---

To listen to the file checkbox

DECLARATION

---

```
protected class OptionsDialog.GUIPanel.FileCheckBoxChangeListener
extends java.lang.Object
implements javax.swing.event.ChangeListener
```

CONSTRUCTORS

---

- *OptionsDialog.GUIPanel.FileCheckBoxChangeListener*  
protected **OptionsDialog.GUIPanel.FileCheckBoxChangeListener**( )

METHODS

---

- *stateChanged*  
public void **stateChanged**( javax.swing.event.ChangeEvent e )

**13.2.28 CLASS OptionsDialog.GUIPanel.InitFileBrowseButtonAction**

---

The action associated with the 'browse' button

DECLARATION

---

```
protected class OptionsDialog.GUIPanel.InitFileBrowseButtonAction
extends javax.swing.AbstractAction
```

CONSTRUCTORS

---

- *OptionsDialog.GUIPanel.InitFileBrowseButtonAction*  
protected **OptionsDialog.GUIPanel.InitFileBrowseButtonAction**( )

METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )



METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
 public synchronized void **addPropertyChangeListener**(  
   java.beans.PropertyChangeListener   )
- *clone*  
 protected Object **clone**( )
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String   , java.lang.Object   ,  
   java.lang.Object   )
- *getKeys*  
 public Object **getKeys**( )
- *getPropertyChangeListeners*  
 public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
 public Object **getValue**( java.lang.String   )
- *isEnabled*  
 public boolean **isEnabled**( )
- *putValue*  
 public void **putValue**( java.lang.String   , java.lang.Object   )
- *readObject*  
 private void **readObject**( java.io.ObjectInputStream   )
- *removePropertyChangeListener*  
 public synchronized void **removePropertyChangeListener**(  
   java.beans.PropertyChangeListener   )
- *setEnabled*  
 public void **setEnabled**( boolean   )
- *writeObject*  
 private void **writeObject**( java.io.ObjectOutputStream   )

**13.2.29 CLASS OptionsDialog.InterpreterPanel**

---

The interpreter option panel

DECLARATION

---

```
protected class OptionsDialog.InterpreterPanel
extends javax.swing.JPanel
```

CONSTRUCTORS

---

- *OptionsDialog.InterpreterPanel*  
 public **OptionsDialog.InterpreterPanel**( )  
   – **Usage**  
     \* Creates a new panel

METHODS

---

- *getFilename*  
public String **getFilename**( )  
  
– **Usage**  
    \* Returns the initialization file name  

---
- *getName*  
public String **getName**( )  
  
– **Usage**  
    \* Returns the name to give to the interpreter  

---
- *isExportationSelected*  
public boolean **isExportationSelected**( )  
  
– **Usage**  
    \* Has the interpreter to be exported?  

---
- *isInitializationSelected*  
public boolean **isInitializationSelected**( )  
  
– **Usage**  
    \* Is the initialization file checkbox selected  

---
- *setExportationSelected*  
public void **setExportationSelected**( boolean b )  
  
– **Usage**  
    \* Sets the state of the checkbox  

---
- *setFilename*  
public void **setFilename**( java.lang.String s )  
  
– **Usage**  
    \* Sets the initialization file name  

---
- *setInitializationSelected*  
public void **setInitializationSelected**( boolean b )  
  
– **Usage**  
    \* Sets the initialization file checkbox state  

---
- *setName*  
public void **setName**( java.lang.String s )  
  
– **Usage**  
    \* Sets the interpreter name

METHODS INHERITED FROM CLASS javax.swing.JPanel

---

- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getUI*  
public PanelUI getUI( )
- *getUIClassID*  
public String getUIClassID( )
- *paramString*  
protected String paramString( )
- *setUI*  
public void setUI( javax.swing.plaf.PanelUI )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

METHODS INHERITED FROM CLASS javax.swing.JComponent

---

- *\_paintImmediately*  
void \_paintImmediately( int , int , int , int )
- *<clinit>*  
static void <clinit>( )
- *addAncestorListener*  
public void addAncestorListener( javax.swing.event.AncestorListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *addVetoableChangeListener*  
public synchronized void addVetoableChangeListener( java.beans.VetoableChangeListener )
- *adjustPaintFlags*  
private void adjustPaintFlags( )
- *alwaysOnTop*  
boolean alwaysOnTop( )
- *checkIfChildObscuredBySibling*  
boolean checkIfChildObscuredBySibling( )
- *componentInputMapChanged*  
void componentInputMapChanged( javax.swing.ComponentInputMap )
- *computeVisibleRect*  
static final void computeVisibleRect( java.awt.Component , java.awt.Rectangle )
- *computeVisibleRect*  
public void computeVisibleRect( java.awt.Rectangle )
- *compWriteObjectNotify*  
void compWriteObjectNotify( )

- *contains*  
public boolean contains( int , int )
- *createToolTip*  
public JToolTip createToolTip( )
- *deregisterNextFocusableComponent*  
private void deregisterNextFocusableComponent( )
- *disable*  
public void disable( )
- *enable*  
public void enable( )
- *enableSerialization*  
void enableSerialization( )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , byte , byte )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , char , char )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getAncestorListeners*  
public AncestorListener getAncestorListeners( )
- *getAutoscrolls*  
public boolean getAutoscrolls( )

- *getBorder*  
public Border **getBorder**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary **getClientProperties**( )
- *getClientProperty*  
public final Object **getClientProperty**( java.lang.Object )
- *getComponentGraphics*  
protected Graphics **getComponentGraphics**( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int **getConditionForKeyStroke**( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean **getCreatedDoubleBuffer**( boolean )
- *getDebugGraphicsOptions*  
public int **getDebugGraphicsOptions**( )
- *getDefaultLocale*  
public static Locale **getDefaultLocale**( )
- *getFlag*  
private boolean **getFlag**( int )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getHeight*  
public int **getHeight**( )
- *getInputMap*  
public final InputMap **getInputMap**( )
- *getInputMap*  
public final InputMap **getInputMap**( int )
- *getInputMap*  
final InputMap **getInputMap**( int , boolean )
- *getInputVerifier*  
public InputVerifier **getInputVerifier**( )
- *getInsets*  
public Insets **getInsets**( )
- *getInsets*  
public Insets **getInsets**( java.awt.Insets )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set **getManagingFocusBackwardTraversalKeys**( )
- *getManagingFocusForwardTraversalKeys*  
static Set **getManagingFocusForwardTraversalKeys**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getNextFocusableComponent*  
public Component **getNextFocusableComponent**( )

- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )
- *getToolTipText*  
public String getToolTipText( )
- *getToolTipText*  
public String getToolTipText( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container getTopLevelAncestor( )
- *getTransferHandler*  
public TransferHandler getTransferHandler( )
- *getUIClassID*  
public String getUIClassID( )
- *getVerifyInputWhenFocusTarget*  
public boolean getVerifyInputWhenFocusTarget( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener getVetoableChangeListeners( )
- *getWidth*  
public int getWidth( )
- *getVisibleRect*  
public Rectangle getVisibleRect( )
- *getWriteObjCounter*  
static byte getWriteObjCounter( javax.swing.JComponent )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *grabFocus*  
public void grabFocus( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component )
- *isManagingFocus*  
public boolean isManagingFocus( )

- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )
- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )

- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke ,  
java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent ,  
java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener(  
java.beans.VetoableChangeListener )
- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )



- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )
- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )

- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )
- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

METHODS INHERITED FROM CLASS `java.awt.Container`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *add*  
`public Component add( java.awt.Component )`
  - *add*  
`public Component add( java.awt.Component , int )`
  - *add*  
`public void add( java.awt.Component , java.lang.Object )`
  - *add*  
`public void add( java.awt.Component , java.lang.Object , int )`
  - *add*  
`public Component add( java.lang.String , java.awt.Component )`
  - *addContainerListener*  
`public synchronized void addContainerListener( java.awt.event.ContainerListener )`
  - *addImpl*  
`protected void addImpl( java.awt.Component , java.lang.Object , int )`
  - *addNotify*  
`public void addNotify( )`
  - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
  - *adjustDecendantsOnParent*  
`void adjustDecendantsOnParent( int )`
  - *adjustDescendants*  
`void adjustDescendants( int )`
  - *adjustListeningChildren*  
`void adjustListeningChildren( long , int )`
  - *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
  - *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
  - *checkGD*  
`void checkGD( java.lang.String )`
  - *clearCurrentFocusCycleRootOnHide*  
`void clearCurrentFocusCycleRootOnHide( )`
  - *clearMostRecentFocusOwnerOnHide*  
`void clearMostRecentFocusOwnerOnHide( )`
  - *containsFocus*  
`final boolean containsFocus( )`
  - *countComponents*  
`public int countComponents( )`
  - *countHierarchyMembers*  
`int countHierarchyMembers( )`
  - *createChildHierarchyEvents*  
`void createChildHierarchyEvents( int , long , boolean )`

- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )

- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseEventTarget*  
Component getMouseEventTarget( int , int , boolean )
- *getMouseEventTarget*  
private Component getMouseEventTarget( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component getMouseEventTargetImpl( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *insets*  
public Insets insets( )
- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )

- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )
- *printComponents*  
public void printComponents( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processContainerEvent*  
protected void processContainerEvent( java.awt.event.ContainerEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *proxyEnableEvents*  
void proxyEnableEvents( long )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public void remove( java.awt.Component )
- *remove*  
public void remove( int )
- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )

- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void setFocusTraversalPolicy( java.awt.FocusTraversalPolicy )
- *setFont*  
public void setFont( java.awt.Font )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setZOrder*  
void setZOrder( java.awt.Component , int )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusDownCycle*  
public void transferFocusDownCycle( )
- *update*  
public void update( java.awt.Graphics )
- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )
- *addInputMethodListener*  
public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )

- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *areInputMethodsEnabled*  
boolean areInputMethodsEnabled( )
- *autoProcessMouseWheel*  
void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )
- *autoTransferFocus*  
final void autoTransferFocus( boolean )
- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )



- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *doAutoTransfer*  
private void doAutoTransfer( boolean )
- *doLayout*  
public void doLayout( )
- *enable*  
public void enable( )
- *enable*  
public void enable( boolean )
- *enableEvents*  
protected final void enableEvents( long )
- *enableInputMethods*  
public void enableInputMethods( boolean )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , int , int )

- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent( )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getBackBuffer*  
Image getBackBuffer( )
- *getBackground*  
public Color getBackground( )
- *getBounds*  
public Rectangle getBounds( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy getBufferStrategy( )
- *getColorModel*  
public ColorModel getColorModel( )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener getComponentListeners( )
- *getComponentOrientation*  
public ComponentOrientation getComponentOrientation( )
- *getCursor*  
public Cursor getCursor( )
- *getDropTarget*  
public synchronized DropTarget getDropTarget( )
- *getFocusCycleRootAncestor*  
public Container getFocusCycleRootAncestor( )
- *getFocusListeners*  
public synchronized FocusListener getFocusListeners( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set getFocusTraversalKeys\_NoIDCheck( int )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalKeysEnabled*  
public boolean getFocusTraversalKeysEnabled( )
- *getFont\_NoClientCode*  
final Font getFont\_NoClientCode( )
- *getFont*  
public Font getFont( )

- *getFontMetrics*  
public FontMetrics getFontMetrics( java.awt.Font    )
- *getForeground*  
public Color getForeground( )
- *getGraphics*  
public Graphics getGraphics( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getHeight*  
public int getHeight( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener getHierarchyBoundsListeners( )
- *getHierarchyListeners*  
public synchronized HierarchyListener getHierarchyListeners( )
- *getIgnoreRepaint*  
public boolean getIgnoreRepaint( )
- *getInputContext*  
public InputContext getInputContext( )
- *getInputMethodListeners*  
public synchronized InputMethodListener getInputMethodListeners( )
- *getInputMethodRequests*  
public InputMethodRequests getInputMethodRequests( )
- *getKeyListeners*  
public synchronized KeyListener getKeyListeners( )
- *getListeners*  
public EventListener getListeners( java.lang.Class    )
- *getLocale*  
public Locale getLocale( )
- *getLocation*  
public Point getLocation( )
- *getLocation*  
public Point getLocation( java.awt.Point    )
- *getLocationOnScreen\_NoTreeLock*  
final Point getLocationOnScreen\_NoTreeLock( )
- *getLocationOnScreen*  
public Point getLocationOnScreen( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseListeners*  
public synchronized MouseListener getMouseListeners( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )

- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )
- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )

- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )

- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )
- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )

- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )

- *removeMouseListener*  
public synchronized void removeMouseListener(  
java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
protected boolean requestFocus( boolean    )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean    , boolean    )
- *requestFocusInWindow*  
public boolean requestFocusInWindow(    )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean    )
- *resetGC*  
void resetGC(    )
- *reshape*  
public void reshape( int    , int    , int    , int    )
- *resize*  
public void resize( java.awt.Dimension    )
- *resize*  
public void resize( int    , int    )
- *setBackground*  
public void setBackground( java.awt.Color    )
- *setBounds*  
public void setBounds( int    , int    , int    , int    )
- *setBounds*  
public void setBounds( java.awt.Rectangle    )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation    )
- *setCursor*  
public void setCursor( java.awt.Cursor    )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget    )
- *setEnabled*  
public void setEnabled( boolean    )



- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusUpCycle*  
public void transferFocusUpCycle( )
- *update*  
public void update( java.awt.Graphics )
- *updateCursorImmediately*  
final void updateCursorImmediately( )
- *validate*  
public void validate( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.30 CLASS *OptionsDialog.InterpreterPanel.CheckBoxChangeListener*

---

To listen to the checkbox

#### DECLARATION

---

```
protected class OptionsDialog.InterpreterPanel.CheckBoxChangeListener
extends java.lang.Object
implements javax.swing.event.ChangeListener
```

#### CONSTRUCTORS

---

- *OptionsDialog.InterpreterPanel.CheckBoxChangeListener*  
`protected OptionsDialog.InterpreterPanel.CheckBoxChangeListener( )`

#### METHODS

---

- *stateChanged*  
`public void stateChanged( javax.swing.event.ChangeEvent e )`

### 13.2.31 CLASS *OptionsDialog.InterpreterPanel.FileCheckBoxChangeListener*

---

To listen to the file checkbox

#### DECLARATION

---

```
protected class OptionsDialog.InterpreterPanel.FileCheckBoxChangeListener
extends java.lang.Object
implements javax.swing.event.ChangeListener
```

#### CONSTRUCTORS

---

- *OptionsDialog.InterpreterPanel.FileCheckBoxChangeListener*  
`protected OptionsDialog.InterpreterPanel.FileCheckBoxChangeListener( )`

#### METHODS

---

- *stateChanged*  
`public void stateChanged( javax.swing.event.ChangeEvent e )`

### 13.2.32 CLASS OptionsDialog.InterpreterPanel.InitFileBrowseButtonAction

---

The action associated with the 'browse' button

#### DECLARATION

---

```
protected class OptionsDialog.InterpreterPanel.InitFileBrowseButtonAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *OptionsDialog.InterpreterPanel.InitFileBrowseButtonAction*  
protected **OptionsDialog.InterpreterPanel.InitFileBrowseButtonAction**( )

#### METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.33 CLASS *OptionsDialog.LPLAddButtonAction*

---

The action associated with the 'add' button of the library path panel

#### DECLARATION

---

```
protected class OptionsDialog.LPLAddButtonAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *OptionsDialog.LPLAddButtonAction*  
protected **OptionsDialog.LPLAddButtonAction**( )

#### METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.34 CLASS OptionsDialog.OKButtonAction

---

The action associated with the 'OK' button

#### DECLARATION

---

```
protected class OptionsDialog.OKButtonAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *OptionsDialog.OKButtonAction*  
protected **OptionsDialog.OKButtonAction**( )

#### METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.35 CLASS OptionsDialog.OptionSet

---

To save the options

#### DECLARATION

---

```
public static class OptionsDialog.OptionSet
extends java.lang.Object
```

#### FIELDS

---

- public String classPath
  - The class path
- public String libraryPath
  - The library path
- public boolean isInterpreterSelected
  - The interpreter checkbox state
- public String interpreterName
  - The interpreter name
- public boolean interpreterFileSelected
  - The interpreter file checkbox state
- public String interpreterFilename
  - The interpreter initialization file name
- public boolean isGUISelected
  - The GUI checkbox state
- public String guiName
  - The GUI name
- public boolean isOutputSelected
  - The output checkbox state
- public boolean isErrorSelected
  - The error checkbox state
- public boolean guiFileSelected
  - The GUI file checkbox state
- public String guiFilename
  - The GUI initialization file name

CONSTRUCTORS

---

- *OptionsDialog.OptionSet*  
**public OptionsDialog.OptionSet( )**  
 – **Usage**  
 \* Creates a new option set with default values
- *OptionsDialog.OptionSet*  
**public OptionsDialog.OptionSet( koala.dynamicjava.gui.OptionsDialog d )**  
 – **Usage**  
 \* Creates a new option set

**13.2.36 CLASS OptionsDialog.UCOKButtonAction**

---

The action associated with the 'OK' button of the URL chooser

DECLARATION

---

```
protected class OptionsDialog.UCOKButtonAction
extends javax.swing.AbstractAction
```

CONSTRUCTORS

---

- *OptionsDialog.UCOKButtonAction*  
**protected OptionsDialog.UCOKButtonAction( )**

METHODS

---

- *actionPerformed*  
**public void actionPerformed( java.awt.event.ActionEvent e )**

METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
**public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener l )**
- *clone*  
**protected Object clone( )**
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
- *getKeys*  
**public Object getKeys( )**

- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object **getValue**( java.lang.String )
- *isEnabled*  
public boolean **isEnabled**( )
- *putValue*  
public void **putValue**( java.lang.String , java.lang.Object )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void **setEnabled**( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.37 CLASS StatusBar

---

The status bar

#### DECLARATION

---

```
public class StatusBar
extends javax.swing.JPanel
implements MessageHandler
```

#### CONSTRUCTORS

---

- *StatusBar*  
**public StatusBar**( koala.dynamicjava.gui.resource.ResourceManager **rm** )
  - **Usage**
    - \* Creates a new status bar
  - **Parameters**
    - \* **rm** - the resource manager that finds the message

#### METHODS

---

- *setLine*  
**public void setLine**( int **n** )
  - **Usage**
    - \* Sets the line number message
  - **Parameters**
    - \* **n** - the line number



- 
- *setMainMessage*  
`public void setMainMessage( java.lang.String s )`
    - **Usage**
      - \* Sets the main message
    - **Parameters**
      - \* **s** - the message
- 
- *setMainMessage*  
`public void setMainMessage( java.lang.String s, java.lang.String s2 )`
    - **Usage**
      - \* Sets the main message to display
    - **Parameters**
      - \* **s** - the message
      - \* **s2** - a string to concatenate with the message
- 
- *setMessage*  
`public void setMessage( java.lang.String s )`
    - **Usage**
      - \* Sets a temporary message
    - **Parameters**
      - \* **s** - the message
- 
- *setMessage*  
`public void setMessage( java.lang.String s, java.lang.String s2 )`
    - **Usage**
      - \* Sets a temporary message to display
    - **Parameters**
      - \* **s** - the message
      - \* **s2** - a string to concatenate with the message

#### METHODS INHERITED FROM CLASS javax.swing.JPanel

---

- *getAccessibleContext*  
`public AccessibleContext getAccessibleContext( )`
- *getUI*  
`public PanelUI getUI( )`
- *getUIClassID*  
`public String getUIClassID( )`
- *paramString*  
`protected String paramString( )`
- *setUI*  
`public void setUI( javax.swing.plaf.PanelUI )`
- *updateUI*  
`public void updateUI( )`
- *writeObject*  
`private void writeObject( java.io.ObjectOutputStream )`

## METHODS INHERITED FROM CLASS javax.swing.JComponent

- 
- *\_paintImmediately*  
void **\_paintImmediately**( int , int , int , int )
  - *<clinit>*  
static void **<clinit>**( )
  - *addAncestorListener*  
public void **addAncestorListener**( javax.swing.event.AncestorListener )
  - *addNotify*  
public void **addNotify**( )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.beans.PropertyChangeListener )
  - *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.lang.String , java.beans.PropertyChangeListener )
  - *addVetoableChangeListener*  
public synchronized void **addVetoableChangeListener**( java.beans.VetoableChangeListener )
  - *adjustPaintFlags*  
private void **adjustPaintFlags**( )
  - *alwaysOnTop*  
boolean **alwaysOnTop**( )
  - *checkIfChildObscuredBySibling*  
boolean **checkIfChildObscuredBySibling**( )
  - *componentInputMapChanged*  
void **componentInputMapChanged**( javax.swing.ComponentInputMap )
  - *computeVisibleRect*  
static final void **computeVisibleRect**( java.awt.Component , java.awt.Rectangle )
  - *computeVisibleRect*  
public void **computeVisibleRect**( java.awt.Rectangle )
  - *compWriteObjectNotify*  
void **compWriteObjectNotify**( )
  - *contains*  
public boolean **contains**( int , int )
  - *createToolTip*  
public JToolTip **createToolTip**( )
  - *deregisterNextFocusableComponent*  
private void **deregisterNextFocusableComponent**( )
  - *disable*  
public void **disable**( )
  - *enable*  
public void **enable**( )
  - *enableSerialization*  
void **enableSerialization**( )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , boolean , boolean )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , byte , byte )
  - *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , char , char )

- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getAncestorListeners*  
public AncestorListener getAncestorListeners( )
- *getAutoscrolls*  
public boolean getAutoscrolls( )
- *getBorder*  
public Border getBorder( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary getClientProperties( )
- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )
- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )

- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )

- *getToolTipText*  
public String getToolTipText( )
- *getToolTipText*  
public String getToolTipText( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container getTopLevelAncestor( )
- *getTransferHandler*  
public TransferHandler getTransferHandler( )
- *getUIClassID*  
public String getUIClassID( )
- *getVerifyInputWhenFocusTarget*  
public boolean getVerifyInputWhenFocusTarget( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener getVetoableChangeListeners( )
- *getWidth*  
public int getWidth( )
- *getVisibleRect*  
public Rectangle getVisibleRect( )
- *getWriteObjCounter*  
static byte getWriteObjCounter( javax.swing.JComponent )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *grabFocus*  
public void grabFocus( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component )
- *isManagingFocus*  
public boolean isManagingFocus( )
- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )

- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )

- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener ,  
java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String ,  
java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener(  
java.beans.VetoableChangeListener )
- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )

- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )



- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )

- *adjustDecendantsOnParent*  
void **adjustDecendantsOnParent**( int    )
- *adjustDescendants*  
void **adjustDescendants**( int    )
- *adjustListeningChildren*  
void **adjustListeningChildren**( long    , int    )
- *applyComponentOrientation*  
public void **applyComponentOrientation**( java.awt.ComponentOrientation    )
- *areFocusTraversalKeysSet*  
public boolean **areFocusTraversalKeysSet**( int    )
- *checkGD*  
void **checkGD**( java.lang.String    )
- *clearCurrentFocusCycleRootOnHide*  
void **clearCurrentFocusCycleRootOnHide**(    )
- *clearMostRecentFocusOwnerOnHide*  
void **clearMostRecentFocusOwnerOnHide**(    )
- *containsFocus*  
final boolean **containsFocus**(    )
- *countComponents*  
public int **countComponents**(    )
- *countHierarchyMembers*  
int **countHierarchyMembers**(    )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int    , long    , boolean    )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int    , java.awt.Component    , java.awt.Container    , long    , boolean    )
- *deliverEvent*  
public void **deliverEvent**( java.awt.Event    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchEventToSelf*  
void **dispatchEventToSelf**( java.awt.AWTEvent    )
- *doLayout*  
public void **doLayout**(    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *findComponentAt*  
public Component **findComponentAt**( int    , int    )
- *findComponentAt*  
final Component **findComponentAt**( int    , int    , boolean    )
- *findComponentAt*  
public Component **findComponentAt**( java.awt.Point    )
- *findTraversalRoot*  
private Container **findTraversalRoot**(    )
- *getAccessibleAt*  
Accessible **getAccessibleAt**( java.awt.Point    )
- *getAccessibleChild*  
Accessible **getAccessibleChild**( int    )
- *getAccessibleChildrenCount*  
int **getAccessibleChildrenCount**(    )

- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getComponent*  
public Component **getComponent**( int )
- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentCount*  
public int **getComponentCount**( )
- *getComponents\_NoClientCode*  
final Component **getComponents\_NoClientCode**( )
- *getComponents*  
public Component **getComponents**( )
- *getContainerListeners*  
public synchronized ContainerListener **getContainerListeners**( )
- *getDropTargetEventTarget*  
Component **getDropTargetEventTarget**( int , int , boolean )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy **getFocusTraversalPolicy**( )
- *getInsets*  
public Insets **getInsets**( )
- *getLayout*  
public LayoutManager **getLayout**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )

- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )

- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )
- *removeAll*  
public void **removeAll**(    )
- *removeContainerListener*  
public synchronized void **removeContainerListener**( java.awt.event.ContainerListener    )
- *removeNotify*  
public void **removeNotify**(    )
- *setFocusCycleRoot*  
public void **setFocusCycleRoot**( boolean    )
- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int    , java.util.Set    )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy    )
- *setFont*  
public void **setFont**( java.awt.Font    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setZOrder*  
void **setZOrder**( java.awt.Component    , int    )
- *transferFocusBackward*  
public void **transferFocusBackward**(    )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**(    )
- *update*  
public void **update**( java.awt.Graphics    )
- *validate*  
public void **validate**(    )
- *validateTree*  
protected void **validateTree**(    )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )

METHODS INHERITED FROM CLASS `java.awt.Component`

- *<clinit>*  
`static void <clinit>( )`
- *action*  
`public boolean action( java.awt.Event , java.lang.Object )`
- *add*  
`public synchronized void add( java.awt.PopupMenu )`
- *addComponentListener*  
`public synchronized void addComponentListener( java.awt.event.ComponentListener )`
- *addFocusListener*  
`public synchronized void addFocusListener( java.awt.event.FocusListener )`
- *addHierarchyBoundsListener*  
`public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )`
- *addHierarchyListener*  
`public void addHierarchyListener( java.awt.event.HierarchyListener )`
- *addInputMethodListener*  
`public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )`
- *addKeyListener*  
`public synchronized void addKeyListener( java.awt.event.KeyListener )`
- *addMouseListener*  
`public synchronized void addMouseListener( java.awt.event.MouseListener )`
- *addMouseMotionListener*  
`public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )`
- *addMouseWheelListener*  
`public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )`
- *addNotify*  
`public void addNotify( )`
- *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
- *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
- *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
- *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
- *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
- *areInputMethodsEnabled*  
`boolean areInputMethodsEnabled( )`
- *autoProcessMouseWheel*  
`void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )`
- *autoTransferFocus*  
`final void autoTransferFocus( boolean )`

- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )

- *dispatchEvent*  
public final void **dispatchEvent**( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean **dispatchMouseWheelToAncestor**( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void **doAutoTransfer**( boolean    )
- *doLayout*  
public void **doLayout**( )
- *enable*  
public void **enable**( )
- *enable*  
public void **enable**( boolean    )
- *enableEvents*  
protected final void **enableEvents**( long    )
- *enableInputMethods*  
public void **enableInputMethods**( boolean    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean **eventTypeEnabled**( int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , int    , int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , java.lang.Object    , java.lang.Object    )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getAccessibleIndexInParent*  
int **getAccessibleIndexInParent**( )
- *getAccessibleStateSet*  
AccessibleStateSet **getAccessibleStateSet**( )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle    )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )



- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )

- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener **getMouseMotionListeners**( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener **getMouseWheelListeners**( )
- *getName*  
public String **getName**( )
- *getNativeContainer*  
Container **getNativeContainer**( )
- *getParent\_NoClientCode*  
final Container **getParent\_NoClientCode**( )
- *getParent*  
public Container **getParent**( )
- *getPeer*  
public ComponentPeer **getPeer**( )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getSize*  
public Dimension **getSize**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getToolkitImpl*  
final Toolkit **getToolkitImpl**( )
- *getTreeLock*  
public final Object **getTreeLock**( )
- *getWidth*  
public int **getWidth**( )

- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *gotFocus*  
public boolean **gotFocus**( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean **handleEvent**( java.awt.Event )
- *hasFocus*  
public boolean **hasFocus**( )
- *hide*  
public void **hide**( )
- *imageUpdate*  
public boolean **imageUpdate**( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *inside*  
public boolean **inside**( int , int )
- *invalidate*  
public void **invalidate**( )
- *isBackgroundSet*  
public boolean **isBackgroundSet**( )
- *isCursorSet*  
public boolean **isCursorSet**( )
- *isDisplayable*  
public boolean **isDisplayable**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isEnabled*  
public boolean **isEnabled**( )
- *isEnabledImpl*  
final boolean **isEnabledImpl**( )
- *isFocusable*  
public boolean **isFocusable**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusOwner*  
public boolean **isFocusOwner**( )
- *isFocusTraversable*  
public boolean **isFocusTraversable**( )
- *isFocusTraversableOverridden*  
final boolean **isFocusTraversableOverridden**( )
- *isFontSet*  
public boolean **isFontSet**( )
- *isForegroundSet*  
public boolean **isForegroundSet**( )
- *isLightweight*  
public boolean **isLightweight**( )

- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )

- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvent*  
boolean postsOldMouseEvent( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )

- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *repaint*  
public void repaint( )
- *repaint*  
public void repaint( int , int , int , int )
- *repaint*  
public void repaint( long )
- *repaint*  
public void repaint( long , int , int , int , int )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )

- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )

- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusUpCycle*  
public void transferFocusUpCycle( )
- *update*  
public void update( java.awt.Graphics )
- *updateCursorImmediately*  
final void updateCursorImmediately( )
- *validate*  
public void validate( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.38 CLASS StatusBar.DisplayThread

---

To display the main message

#### DECLARATION

---

```
protected class StatusBar.DisplayThread
extends java.lang.Thread
```

#### CONSTRUCTORS

---

- *StatusBar.DisplayThread*  
**public StatusBar.DisplayThread( )**

#### METHODS

---

- *run*  
**public void run( )**



METHODS INHERITED FROM CLASS java.lang.Thread

---

- *<clinit>*  
static void <clinit>( )
- *activeCount*  
public static int activeCount( )
- *blockedOn*  
private void blockedOn( sun.nio.ch.Interruptible )
- *checkAccess*  
public final void checkAccess( )
- *countStackFrames*  
public native int countStackFrames( )
- *currentThread*  
public static native Thread currentThread( )
- *destroy*  
public void destroy( )
- *dumpStack*  
public static void dumpStack( )
- *enumerate*  
public static int enumerate( java.lang.Thread [] )
- *exit*  
private void exit( )
- *getContextClassLoader*  
public ClassLoader getContextClassLoader( )
- *getName*  
public final String getName( )
- *getPriority*  
public final int getPriority( )
- *getThreadGroup*  
public final ThreadGroup getThreadGroup( )
- *holdsLock*  
public static native boolean holdsLock( java.lang.Object )
- *init*  
private void init( java.lang.ThreadGroup , java.lang.Runnable , java.lang.String , long )
- *interrupt*  
public void interrupt( )
- *interrupt0*  
private native void interrupt0( )
- *interrupted*  
public static boolean interrupted( )
- *isAlive*  
public final native boolean isAlive( )
- *isDaemon*  
public final boolean isDaemon( )
- *isInterrupted*  
public boolean isInterrupted( )
- *isInterrupted*  
private native boolean isInterrupted( boolean )

- *join*  
public final void join( )
- *join*  
public final synchronized void join( long )
- *join*  
public final synchronized void join( long , int )
- *nextThreadNum*  
private static synchronized int nextThreadNum( )
- *registerNatives*  
private static native void registerNatives( )
- *resume*  
public final void resume( )
- *resume0*  
private native void resume0( )
- *run*  
public void run( )
- *setContextClassLoader*  
public void setContextClassLoader( java.lang.ClassLoader )
- *setDaemon*  
public final void setDaemon( boolean )
- *setName*  
public final void setName( java.lang.String )
- *setPriority*  
public final void setPriority( int )
- *setPriority0*  
private native void setPriority0( int )
- *sleep*  
public static native void sleep( long )
- *sleep*  
public static void sleep( long , int )
- *start*  
public synchronized native void start( )
- *stop*  
public final void stop( )
- *stop*  
public final synchronized void stop( java.lang.Throwable )
- *stop0*  
private native void stop0( java.lang.Object )
- *suspend*  
public final void suspend( )
- *suspend0*  
private native void suspend0( )
- *toString*  
public String toString( )
- *yield*  
public static native void yield( )

### 13.2.39 CLASS *StringList*

---

This component is used to manipulate a list of strings

DECLARATION

---

```
public class StringList
extends javax.swing.JPanel
implements koala.dynamicjava.gui.resource.ActionMap
```

CONSTRUCTORS

---

- *StringList*  
**public StringList( javax.swing.Action addAction )**
  - **Usage**
    - \* Creates a new list
  - **Parameters**
    - \* **addAction** - the action associated with the add button

METHODS

---

- *add*  
**public void add( java.lang.String s )**
  - **Usage**
    - \* Adds a string
- *getAction*  
**public Action getAction( java.lang.String key )**
  - **Usage**
    - \* Returns the action associated with the given string or null on error
  - **Parameters**
    - \* **key** - the key mapped with the action to get
  - **Exceptions**
    - \* **koala.dynamicjava.gui.resource.MissingListenerException** - if the action is not found
- *getStrings*  
**public String getStrings( )**
  - **Usage**
    - \* Returns the strings contained in the list
- *setStrings*  
**public void setStrings( java.lang.String [] strings )**
  - **Usage**
    - \* Sets the strings

- *updateButtons*  
protected void **updateButtons**( )
- **Usage**  
\* Updates the state of the buttons

---

#### METHODS INHERITED FROM CLASS javax.swing.JPanel

---

- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getUI*  
public PanelUI **getUI**( )
- *getUIClassID*  
public String **getUIClassID**( )
- *paramString*  
protected String **paramString**( )
- *setUI*  
public void **setUI**( javax.swing.plaf.PanelUI )
- *updateUI*  
public void **updateUI**( )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

---

#### METHODS INHERITED FROM CLASS javax.swing.JComponent

---

- *\_paintImmediately*  
void **\_paintImmediately**( int , int , int , int )
- *<clinit>*  
static void **<clinit>**( )
- *addAncestorListener*  
public void **addAncestorListener**( javax.swing.event.AncestorListener )
- *addNotify*  
public void **addNotify**( )
- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**( java.lang.String , java.beans.PropertyChangeListener )
- *addVetoableChangeListener*  
public synchronized void **addVetoableChangeListener**( java.beans.VetoableChangeListener )
- *adjustPaintFlags*  
private void **adjustPaintFlags**( )
- *alwaysOnTop*  
boolean **alwaysOnTop**( )
- *checkIfChildObscuredBySibling*  
boolean **checkIfChildObscuredBySibling**( )
- *componentInputMapChanged*  
void **componentInputMapChanged**( javax.swing.ComponentInputMap )

- *computeVisibleRect*  
static final void computeVisibleRect( java.awt.Component , java.awt.Rectangle )
- *computeVisibleRect*  
public void computeVisibleRect( java.awt.Rectangle )
- *compWriteObjectNotify*  
void compWriteObjectNotify( )
- *contains*  
public boolean contains( int , int )
- *createToolTip*  
public JToolTip createToolTip( )
- *deregisterNextFocusableComponent*  
private void deregisterNextFocusableComponent( )
- *disable*  
public void disable( )
- *enable*  
public void enable( )
- *enableSerialization*  
void enableSerialization( )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , byte , byte )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , char , char )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )

- *getAlignmentY*  
public float **getAlignmentY**( )
- *getAncestorListeners*  
public AncestorListener **getAncestorListeners**( )
- *getAutoscrolls*  
public boolean **getAutoscrolls**( )
- *getBorder*  
public Border **getBorder**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary **getClientProperties**( )
- *getClientProperty*  
public final Object **getClientProperty**( java.lang.Object )
- *getComponentGraphics*  
protected Graphics **getComponentGraphics**( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int **getConditionForKeyStroke**( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean **getCreatedDoubleBuffer**( boolean )
- *getDebugGraphicsOptions*  
public int **getDebugGraphicsOptions**( )
- *getDefaultLocale*  
public static Locale **getDefaultLocale**( )
- *getFlag*  
private boolean **getFlag**( int )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getHeight*  
public int **getHeight**( )
- *getInputMap*  
public final InputMap **getInputMap**( )
- *getInputMap*  
public final InputMap **getInputMap**( int )
- *getInputMap*  
final InputMap **getInputMap**( int , boolean )
- *getInputVerifier*  
public InputVerifier **getInputVerifier**( )
- *getInsets*  
public Insets **getInsets**( )
- *getInsets*  
public Insets **getInsets**( java.awt.Insets )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set **getManagingFocusBackwardTraversalKeys**( )
- *getManagingFocusForwardTraversalKeys*  
static Set **getManagingFocusForwardTraversalKeys**( )

- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke getRegisteredKeyStrokes( )
- *getRootPane*  
public JRootPane getRootPane( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean getSuppressDropTarget( )
- *getToolTipLocation*  
public Point getToolTipLocation( java.awt.event.MouseEvent )
- *getToolTipText*  
public String getToolTipText( )
- *getToolTipText*  
public String getToolTipText( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container getTopLevelAncestor( )
- *getTransferHandler*  
public TransferHandler getTransferHandler( )
- *getUIClassID*  
public String getUIClassID( )
- *getVerifyInputWhenFocusTarget*  
public boolean getVerifyInputWhenFocusTarget( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener getVetoableChangeListeners( )
- *getWidth*  
public int getWidth( )
- *getVisibleRect*  
public Rectangle getVisibleRect( )
- *getWriteObjCounter*  
static byte getWriteObjCounter( javax.swing.JComponent )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *grabFocus*  
public void grabFocus( )

- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isLightweightComponent*  
public static boolean isLightweightComponent( java.awt.Component )
- *isManagingFocus*  
public boolean isManagingFocus( )
- *isMaximumSizeSet*  
public boolean isMaximumSizeSet( )
- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )
- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )



- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener , javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener , java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener( java.beans.VetoableChangeListener )

- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )
- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )

- *setForeground*  
public void setForeground( java.awt.Color    )
- *setInputMap*  
public final void setInputMap( int    , javax.swing.InputMap    )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier    )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension    )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension    )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component    )
- *setOpaque*  
public void setOpaque( boolean    )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component    )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension    )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean    )
- *setToolTipText*  
public void setToolTipText( java.lang.String    )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler    )
- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI    )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean    )
- *setVisible*  
public void setVisible( boolean    )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent    , byte    )
- *shouldDebugGraphics*  
int shouldDebugGraphics(    )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent    )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke    )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager(    )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke    )
- *update*  
public void update( java.awt.Graphics    )
- *updateUI*  
public void updateUI(    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

METHODS INHERITED FROM CLASS `java.awt.Container`

- *<clinit>*  
`static void <clinit>( )`
- *add*  
`public Component add( java.awt.Component )`
- *add*  
`public Component add( java.awt.Component , int )`
- *add*  
`public void add( java.awt.Component , java.lang.Object )`
- *add*  
`public void add( java.awt.Component , java.lang.Object , int )`
- *add*  
`public Component add( java.lang.String , java.awt.Component )`
- *addContainerListener*  
`public synchronized void addContainerListener( java.awt.event.ContainerListener )`
- *addImpl*  
`protected void addImpl( java.awt.Component , java.lang.Object , int )`
- *addNotify*  
`public void addNotify( )`
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener )`
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
- *adjustDecendantsOnParent*  
`void adjustDecendantsOnParent( int )`
- *adjustDescendants*  
`void adjustDescendants( int )`
- *adjustListeningChildren*  
`void adjustListeningChildren( long , int )`
- *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
- *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
- *checkGD*  
`void checkGD( java.lang.String )`
- *clearCurrentFocusCycleRootOnHide*  
`void clearCurrentFocusCycleRootOnHide( )`
- *clearMostRecentFocusOwnerOnHide*  
`void clearMostRecentFocusOwnerOnHide( )`
- *containsFocus*  
`final boolean containsFocus( )`
- *countComponents*  
`public int countComponents( )`
- *countHierarchyMembers*  
`int countHierarchyMembers( )`
- *createChildHierarchyEvents*  
`void createChildHierarchyEvents( int , long , boolean )`

- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )

- *getInsets*  
public Insets **getInsets**( )
- *getLayout*  
public LayoutManager **getLayout**( )
- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )
- *invalidate*  
public void **invalidate**( )
- *invalidateTree*  
void **invalidateTree**( )
- *isAncestorOf*  
public boolean **isAncestorOf**( java.awt.Component )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean **isFocusTraversalPolicySet**( )
- *isParentOf*  
boolean **isParentOf**( java.awt.Component )
- *layout*  
public void **layout**( )
- *lightweightPaint*  
void **lightweightPaint**( java.awt.Graphics )
- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics )
- *list*  
public void **list**( java.io.PrintStream , int )
- *list*  
public void **list**( java.io.PrintWriter , int )

- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )
- *printComponents*  
public void printComponents( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processContainerEvent*  
protected void processContainerEvent( java.awt.event.ContainerEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *proxyEnableEvents*  
void proxyEnableEvents( long )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public void remove( java.awt.Component )
- *remove*  
public void remove( int )
- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )

- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy )
- *setFont*  
public void **setFont**( java.awt.Font )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager )
- *setZOrder*  
void **setZOrder**( java.awt.Component , int )
- *transferFocusBackward*  
public void **transferFocusBackward**( )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**( )
- *update*  
public void **update**( java.awt.Graphics )
- *validate*  
public void **validate**( )
- *validateTree*  
protected void **validateTree**( )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS `java.awt.Component`

---

- *<clinit>*  
static void **<clinit>**( )
- *action*  
public boolean **action**( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void **add**( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void **addComponentListener**( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void **addFocusListener**( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void **addHierarchyBoundsListener**( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void **addHierarchyListener**( java.awt.event.HierarchyListener )
- *addInputMethodListener*  
public synchronized void **addInputMethodListener**( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void **addKeyListener**( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void **addMouseListener**( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void **addMouseMotionListener**( java.awt.event.MouseMotionListener )



- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *areInputMethodsEnabled*  
boolean areInputMethodsEnabled( )
- *autoProcessMouseWheel*  
void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )
- *autoTransferFocus*  
final void autoTransferFocus( boolean )
- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )

- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *doAutoTransfer*  
private void doAutoTransfer( boolean )
- *doLayout*  
public void doLayout( )
- *enable*  
public void enable( )
- *enable*  
public void enable( boolean )
- *enableEvents*  
protected final void enableEvents( long )
- *enableInputMethods*  
public void enableInputMethods( boolean )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , int , int )

- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent( )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getBackBuffer*  
Image getBackBuffer( )
- *getBackground*  
public Color getBackground( )
- *getBounds*  
public Rectangle getBounds( )
- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy getBufferStrategy( )
- *getColorModel*  
public ColorModel getColorModel( )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener getComponentListeners( )
- *getComponentOrientation*  
public ComponentOrientation getComponentOrientation( )
- *getCursor*  
public Cursor getCursor( )
- *getDropTarget*  
public synchronized DropTarget getDropTarget( )
- *getFocusCycleRootAncestor*  
public Container getFocusCycleRootAncestor( )
- *getFocusListeners*  
public synchronized FocusListener getFocusListeners( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set getFocusTraversalKeys\_NoIDCheck( int )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalKeysEnabled*  
public boolean getFocusTraversalKeysEnabled( )
- *getFont\_NoClientCode*  
final Font getFont\_NoClientCode( )
- *getFont*  
public Font getFont( )

- *getFontMetrics*  
public FontMetrics getFontMetrics( java.awt.Font    )
- *getForeground*  
public Color getForeground(    )
- *getGraphics*  
public Graphics getGraphics(    )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration(    )
- *getHeight*  
public int getHeight(    )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener getHierarchyBoundsListeners(    )
- *getHierarchyListeners*  
public synchronized HierarchyListener getHierarchyListeners(    )
- *getIgnoreRepaint*  
public boolean getIgnoreRepaint(    )
- *getInputContext*  
public InputContext getInputContext(    )
- *getInputMethodListeners*  
public synchronized InputMethodListener getInputMethodListeners(    )
- *getInputMethodRequests*  
public InputMethodRequests getInputMethodRequests(    )
- *getKeyListeners*  
public synchronized KeyListener getKeyListeners(    )
- *getListeners*  
public EventListener getListeners( java.lang.Class    )
- *getLocale*  
public Locale getLocale(    )
- *getLocation*  
public Point getLocation(    )
- *getLocation*  
public Point getLocation( java.awt.Point    )
- *getLocationOnScreen\_NoTreeLock*  
final Point getLocationOnScreen\_NoTreeLock(    )
- *getLocationOnScreen*  
public Point getLocationOnScreen(    )
- *getMaximumSize*  
public Dimension getMaximumSize(    )
- *getMinimumSize*  
public Dimension getMinimumSize(    )
- *getMouseListeners*  
public synchronized MouseListener getMouseListeners(    )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners(    )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners(    )
- *getName*  
public String getName(    )
- *getNativeContainer*  
Container getNativeContainer(    )

- *getParent\_NoClientCode*  
final Container **getParent\_NoClientCode**( )
- *getParent*  
public Container **getParent**( )
- *getPeer*  
public ComponentPeer **getPeer**( )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getSize*  
public Dimension **getSize**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getToolkitImpl*  
final Toolkit **getToolkitImpl**( )
- *getTreeLock*  
public final Object **getTreeLock**( )
- *getWidth*  
public int **getWidth**( )
- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *gotFocus*  
public boolean **gotFocus**( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean **handleEvent**( java.awt.Event )
- *hasFocus*  
public boolean **hasFocus**( )
- *hide*  
public void **hide**( )
- *imageUpdate*  
public boolean **imageUpdate**( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *inside*  
public boolean **inside**( int , int )
- *invalidate*  
public void **invalidate**( )
- *isBackgroundSet*  
public boolean **isBackgroundSet**( )
- *isCursorSet*  
public boolean **isCursorSet**( )

- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )

- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )
- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )

- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )



- *removeMouseListener*  
public synchronized void removeMouseListener(  
java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
protected boolean requestFocus( boolean    )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean    , boolean    )
- *requestFocusInWindow*  
public boolean requestFocusInWindow(    )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean    )
- *resetGC*  
void resetGC(    )
- *reshape*  
public void reshape( int    , int    , int    , int    )
- *resize*  
public void resize( java.awt.Dimension    )
- *resize*  
public void resize( int    , int    )
- *setBackground*  
public void setBackground( java.awt.Color    )
- *setBounds*  
public void setBounds( int    , int    , int    , int    )
- *setBounds*  
public void setBounds( java.awt.Rectangle    )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation    )
- *setCursor*  
public void setCursor( java.awt.Cursor    )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget    )
- *setEnabled*  
public void setEnabled( boolean    )

- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusUpCycle*  
public void transferFocusUpCycle( )
- *update*  
public void update( java.awt.Graphics )
- *updateCursorImmediately*  
final void updateCursorImmediately( )
- *validate*  
public void validate( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 13.2.40 CLASS **StringList.DownButtonAction**

---

The action associated with the 'down' button

#### DECLARATION

---

```
protected class StringList.DownButtonAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *StringList.DownButtonAction*  
protected **StringList.DownButtonAction**( )

#### METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

#### METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

### 13.2.41 CLASS **StringList.ListSelectionAdapter**

---

To manage selection modifications

#### DECLARATION

---

```
protected class StringList.ListSelectionAdapter
extends java.lang.Object
implements javax.swing.event.ListSelectionListener
```

#### CONSTRUCTORS

---

- *StringList.ListSelectionAdapter*  
`protected StringList.ListSelectionAdapter( )`

#### METHODS

---

- *valueChanged*  
`public void valueChanged( javax.swing.event.ListSelectionEvent e )`

### 13.2.42 CLASS **StringList.RemoveButtonAction**

---

The action associated with the 'remove' button

#### DECLARATION

---

```
protected class StringList.RemoveButtonAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *StringList.RemoveButtonAction*  
`protected StringList.RemoveButtonAction( )`

#### METHODS

---

- *actionPerformed*  
`public void actionPerformed( java.awt.event.ActionEvent e )`

METHODS INHERITED FROM CLASS `javax.swing.AbstractAction`

- 
- *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener(  
java.beans.PropertyChangeListener    )`
  - *clone*  
`protected Object clone( )`
  - *firePropertyChange*  
`protected void firePropertyChange( java.lang.String   , java.lang.Object   ,  
java.lang.Object    )`
  - *getKeys*  
`public Object getKeys( )`
  - *getPropertyChangeListeners*  
`public synchronized PropertyChangeListener getPropertyChangeListeners( )`
  - *getValue*  
`public Object getValue( java.lang.String    )`
  - *isEnabled*  
`public boolean isEnabled( )`
  - *putValue*  
`public void putValue( java.lang.String   , java.lang.Object    )`
  - *readObject*  
`private void readObject( java.io.ObjectInputStream    )`
  - *removePropertyChangeListener*  
`public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )`
  - *setEnabled*  
`public void setEnabled( boolean    )`
  - *writeObject*  
`private void writeObject( java.io.ObjectOutputStream    )`

**13.2.43 CLASS *StringList.UpButtonAction***


---

The action associated with the 'up' button

## DECLARATION

---

protected class <i>StringList.UpButtonAction</i> <b>extends</b> <code>javax.swing.AbstractAction</code>
--

## CONSTRUCTORS

- 
- *StringList.UpButtonAction*  
`protected StringList.UpButtonAction( )`

## METHODS

- 
- *actionPerformed*  
`public void actionPerformed( java.awt.event.ActionEvent   e )`

METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener   )
- *clone*  
protected Object clone(   )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String   , java.lang.Object   ,   
java.lang.Object   )
- *getKeys*  
public Object getKeys(   )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(   )
- *getValue*  
public Object getValue( java.lang.String   )
- *isEnabled*  
public boolean isEnabled(   )
- *putValue*  
public void putValue( java.lang.String   , java.lang.Object   )
- *readObject*  
private void readObject( java.io.ObjectInputStream   )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener   )
- *setEnabled*  
public void setEnabled( boolean   )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream   )

**13.2.44 CLASS URLChooser**

---

A component used to enter an URL or to choose a local file

DECLARATION

---

```
public class URLChooser
extends javax.swing.JDialog
implements koala.dynamicjava.gui.resource.ActionMap
```

CONSTRUCTORS

---

- *URLChooser*  
public **URLChooser**( javax.swing.JDialog   d, javax.swing.Action   okAction )

## – Usage

- \* Creates a new URLChooser

---

– **Parameters**

- \* **d** - the parent dialog
  - \* **okAction** - the action to associate to the ok button
- 

• *URLChooser*

**public URLChooser( javax.swing.JFrame f, javax.swing.Action okAction )**

– **Usage**

- \* Creates a new URLChooser

– **Parameters**

- \* **f** - the parent frame
- \* **okAction** - the action to associate to the ok button

---

## METHODS

---

• *createButtonsPanel*

**protected JPanel createButtonsPanel( )**

– **Usage**

- \* Creates the buttons panel
- 

• *createURLSelectionPanel*

**protected JPanel createURLSelectionPanel( )**

– **Usage**

- \* Creates the URL selection panel
- 

• *getAction*

**public Action getAction( java.lang.String key )**

– **Usage**

- \* Returns the action associated with the given string or null on error

– **Parameters**

- \* **key** - the key mapped with the action to get

– **Exceptions**

- \* `koala.dynamicjava.gui.resource.MissingListenerException` - if the action is not found
- 

• *getText*

**public String getText( )**

– **Usage**

- \* Returns the text contained in the text field
- 

• *initialize*

**protected void initialize( javax.swing.Action okAction )**

– **Usage**

- \* Initializes the dialog
-

- *setConstraintsCoords*  
 protected static void **setConstraintsCoords**( java.awt.GridBagConstraints  
**constraints**, int **x**, int **y**, int **width**, int **height** )  
 – **Usage**  
 \* An utility funtion  


---
- *updateClearButtonAction*  
 protected void **updateClearButtonAction**( )  
 – **Usage**  
 \* To update the state of the Clear button  


---
- *updateOKButtonAction*  
 protected void **updateOKButtonAction**( )  
 – **Usage**  
 \* To update the state of the OK button

#### METHODS INHERITED FROM CLASS javax.swing.JDialog

---

- *<clinit>*  
 static void **<clinit>**( )  


---
- *addImpl*  
 protected void **addImpl**( java.awt.Component , java.lang.Object , int )  


---
- *createRootPane*  
 protected JRootPane **createRootPane**( )  


---
- *createRootPaneException*  
 private Error **createRootPaneException**( java.lang.String )  


---
- *dialogInit*  
 protected void **dialogInit**( )  


---
- *getAccessibleContext*  
 public AccessibleContext **getAccessibleContext**( )  


---
- *getContentPane*  
 public Container **getContentPane**( )  


---
- *getDefaultCloseOperation*  
 public int **getDefaultCloseOperation**( )  


---
- *getGlassPane*  
 public Component **getGlassPane**( )  


---
- *getJMenuBar*  
 public JMenuBar **getJMenuBar**( )  


---
- *getLayeredPane*  
 public JLayeredPane **getLayeredPane**( )  


---
- *getRootPane*  
 public JRootPane **getRootPane**( )  


---
- *isDefaultLookAndFeelDecorated*  
 public static boolean **isDefaultLookAndFeelDecorated**( )  


---
- *isRootPaneCheckingEnabled*  
 protected boolean **isRootPaneCheckingEnabled**( )  


---
- *paramString*  
 protected String **paramString**( )  


---



- *processWindowEvent*  
protected void processWindowEvent( java.awt.event.WindowEvent    )
- *remove*  
public void remove( java.awt.Component    )
- *setContentPane*  
public void setContentPane( java.awt.Container    )
- *setDefaultCloseOperation*  
public void setDefaultCloseOperation( int    )
- *setDefaultLookAndFeelDecorated*  
public static void setDefaultLookAndFeelDecorated( boolean    )
- *setGlassPane*  
public void setGlassPane( java.awt.Component    )
- *setJMenuBar*  
public void setJMenuBar( javax.swing.JMenuBar    )
- *setLayeredPane*  
public void setLayeredPane( javax.swing.JLayeredPane    )
- *setLayout*  
public void setLayout( java.awt.LayoutManager    )
- *setRootPane*  
protected void setRootPane( javax.swing.JRootPane    )
- *setRootPaneCheckingEnabled*  
protected void setRootPaneCheckingEnabled( boolean    )
- *update*  
public void update( java.awt.Graphics    )

#### METHODS INHERITED FROM CLASS java.awt.Dialog

---

- *<clinit>*  
static void <clinit>( )
- *addNotify*  
public void addNotify( )
- *conditionalShow*  
private boolean conditionalShow( )
- *constructComponentName*  
String constructComponentName( )
- *dispose*  
public void dispose( )
- *disposeImpl*  
private void disposeImpl( )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getTitle*  
public String getTitle( )
- *hide*  
public void hide( )
- *hideAndDisposeHandler*  
private void hideAndDisposeHandler( )
- *initIDs*  
private static native void initIDs( )

- *interruptBlocking*  
void **interruptBlocking**( )
- *isModal*  
public boolean **isModal**( )
- *isResizable*  
public boolean **isResizable**( )
- *isUndecorated*  
public boolean **isUndecorated**( )
- *paramString*  
protected String **paramString**( )
- *setModal*  
public void **setModal**( boolean )
- *setResizable*  
public void **setResizable**( boolean )
- *setTitle*  
public void **setTitle**( java.lang.String )
- *setUndecorated*  
public void **setUndecorated**( boolean )
- *show*  
public void **show**( )

METHODS INHERITED FROM CLASS `java.awt.Window`

---

- *<clinit>*  
static void **<clinit>**( )
- *addNotify*  
public void **addNotify**( )
- *addOwnedWindow*  
void **addOwnedWindow**( java.lang.ref.WeakReference )
- *addPropertyChangeListener*  
public void **addPropertyChangeListener**( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void **addPropertyChangeListener**( java.lang.String ,  
java.beans.PropertyChangeListener )
- *addWindowFocusListener*  
public synchronized void **addWindowFocusListener**(  
java.awt.event.WindowFocusListener )
- *addWindowListener*  
public synchronized void **addWindowListener**( java.awt.event.WindowListener )
- *addWindowStateListener*  
public synchronized void **addWindowStateListener**(  
java.awt.event.WindowStateListener )
- *adjustDecendantsOnParent*  
void **adjustDecendantsOnParent**( int )
- *adjustListeningChildrenOnParent*  
void **adjustListeningChildrenOnParent**( long , int )
- *applyResourceBundle*  
public void **applyResourceBundle**( java.util.ResourceBundle )
- *applyResourceBundle*  
public void **applyResourceBundle**( java.lang.String )

- *clearMostRecentFocusOwnerOnHide*  
final void clearMostRecentFocusOwnerOnHide( )
- *connectOwnedWindow*  
void connectOwnedWindow( java.awt.Window )
- *constructComponentName*  
String constructComponentName( )
- *createBufferStrategy*  
public void createBufferStrategy( int )
- *createBufferStrategy*  
public void createBufferStrategy( int , java.awt.BufferCapabilities )
- *deliverMouseWheelToAncestor*  
void deliverMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *dispose*  
public void dispose( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *finalize*  
protected void finalize( )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getBufferStrategy*  
public BufferStrategy getBufferStrategy( )
- *getFocusableWindowState*  
public boolean getFocusableWindowState( )
- *getFocusCycleRootAncestor*  
public final Container getFocusCycleRootAncestor( )
- *getFocusOwner*  
public Component getFocusOwner( )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getInputContext*  
public InputContext getInputContext( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocale*  
public Locale getLocale( )
- *getMostRecentFocusOwner*  
public Component getMostRecentFocusOwner( )
- *getOwnedWindows*  
public Window getOwnedWindows( )
- *getOwner*  
public Window getOwner( )
- *getTemporaryLostComponent*  
Component getTemporaryLostComponent( )

- *getToolkit*  
public Toolkit getToolkit( )
- *getWarningString*  
public final String getWarningString( )
- *getWindowFocusListeners*  
public synchronized WindowFocusListener getWindowFocusListeners( )
- *getWindowListeners*  
public synchronized WindowListener getWindowListeners( )
- *getWindowStateListeners*  
public synchronized WindowStateListener getWindowStateListeners( )
- *hide*  
public void hide( )
- *init*  
private void init( java.awt.GraphicsConfiguration )
- *initIDs*  
private static native void initIDs( )
- *isActive*  
public boolean isActive( )
- *isFocusableWindow*  
public final boolean isFocusableWindow( )
- *isFocusCycleRoot*  
public final boolean isFocusCycleRoot( )
- *isFocused*  
public boolean isFocused( )
- *isShowing*  
public boolean isShowing( )
- *ownedInit*  
private void ownedInit( java.awt.Window )
- *pack*  
public void pack( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postProcessKeyEvent*  
void postProcessKeyEvent( java.awt.event.KeyEvent )
- *postWindowEvent*  
synchronized void postWindowEvent( int )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processWindowEvent*  
protected void processWindowEvent( java.awt.event.WindowEvent )
- *processWindowFocusEvent*  
protected void processWindowFocusEvent( java.awt.event.WindowEvent )
- *processWindowStateEvent*  
protected void processWindowStateEvent( java.awt.event.WindowEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removeOwnedWindow*  
void removeOwnedWindow( java.lang.ref.WeakReference )

- *removeWindowFocusListener*  
public synchronized void removeWindowFocusListener( java.awt.event.WindowFocusListener )
- *removeWindowListener*  
public synchronized void removeWindowListener( java.awt.event.WindowListener )
- *removeWindowStateListener*  
public synchronized void removeWindowStateListener( java.awt.event.WindowStateListener )
- *resetGC*  
void resetGC( )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setFocusableWindowState*  
public void setFocusableWindowState( boolean )
- *setFocusCycleRoot*  
public final void setFocusCycleRoot( boolean )
- *setLocationRelativeTo*  
public void setLocationRelativeTo( java.awt.Component )
- *setTemporaryLostComponent*  
Component setTemporaryLostComponent( java.awt.Component )
- *setWarningString*  
private void setWarningString( )
- *show*  
public void show( )
- *toBack*  
public void toBack( )
- *toFront*  
public void toFront( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )
- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )

- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int )
- *adjustDescendants*  
void adjustDescendants( int )
- *adjustListeningChildren*  
void adjustListeningChildren( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *checkGD*  
void checkGD( java.lang.String )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *containsFocus*  
final boolean containsFocus( )
- *countComponents*  
public int countComponents( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )

- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )
- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseEventTarget*  
Component getMouseEventTarget( int , int , boolean )
- *getMouseEventTarget*  
private Component getMouseEventTarget( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component getMouseEventTargetImpl( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )

- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )
- *invalidate*  
public void **invalidate**( )
- *invalidateTree*  
void **invalidateTree**( )
- *isAncestorOf*  
public boolean **isAncestorOf**( java.awt.Component )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean **isFocusTraversalPolicySet**( )
- *isParentOf*  
boolean **isParentOf**( java.awt.Component )
- *layout*  
public void **layout**( )
- *lightweightPaint*  
void **lightweightPaint**( java.awt.Graphics )
- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics )
- *list*  
public void **list**( java.io.PrintStream , int )
- *list*  
public void **list**( java.io.PrintWriter , int )
- *locate*  
public Component **locate**( int , int )
- *minimumSize*  
public Dimension **minimumSize**( )
- *nextFocusHelper*  
boolean **nextFocusHelper**( )
- *numListening*  
int **numListening**( long )
- *paint*  
public void **paint**( java.awt.Graphics )
- *paintComponents*  
public void **paintComponents**( java.awt.Graphics )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics )
- *paramString*  
protected String **paramString**( )
- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent )



- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *preProcessKeyEvent*  
void preProcessKeyEvent( java.awt.event.KeyEvent )
- *print*  
public void print( java.awt.Graphics )
- *printComponents*  
public void printComponents( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processContainerEvent*  
protected void processContainerEvent( java.awt.event.ContainerEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *proxyEnableEvents*  
void proxyEnableEvents( long )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public void remove( java.awt.Component )
- *remove*  
public void remove( int )
- *removeAll*  
public void removeAll( )
- *removeContainerListener*  
public synchronized void removeContainerListener( java.awt.event.ContainerListener )
- *removeNotify*  
public void removeNotify( )
- *setFocusCycleRoot*  
public void setFocusCycleRoot( boolean )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalPolicy*  
public void setFocusTraversalPolicy( java.awt.FocusTraversalPolicy )
- *setFont*  
public void setFont( java.awt.Font )
- *setLayout*  
public void setLayout( java.awt.LayoutManager )
- *setZOrder*  
void setZOrder( java.awt.Component , int )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusDownCycle*  
public void transferFocusDownCycle( )
- *update*  
public void update( java.awt.Graphics )
- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

METHODS INHERITED FROM CLASS `java.awt.Component`

- 
- *<clinit>*  
`static void <clinit>( )`
  - *action*  
`public boolean action( java.awt.Event , java.lang.Object )`
  - *add*  
`public synchronized void add( java.awt.PopupMenu )`
  - *addComponentListener*  
`public synchronized void addComponentListener( java.awt.event.ComponentListener )`
  - *addFocusListener*  
`public synchronized void addFocusListener( java.awt.event.FocusListener )`
  - *addHierarchyBoundsListener*  
`public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )`
  - *addHierarchyListener*  
`public void addHierarchyListener( java.awt.event.HierarchyListener )`
  - *addInputMethodListener*  
`public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )`
  - *addKeyListener*  
`public synchronized void addKeyListener( java.awt.event.KeyListener )`
  - *addMouseListener*  
`public synchronized void addMouseListener( java.awt.event.MouseListener )`
  - *addMouseMotionListener*  
`public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )`
  - *addMouseWheelListener*  
`public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )`
  - *addNotify*  
`public void addNotify( )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
  - *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
  - *adjustListeningChildrenOnParent*  
`void adjustListeningChildrenOnParent( long , int )`
  - *applyComponentOrientation*  
`public void applyComponentOrientation( java.awt.ComponentOrientation )`
  - *areFocusTraversalKeysSet*  
`public boolean areFocusTraversalKeysSet( int )`
  - *areInputMethodsEnabled*  
`boolean areInputMethodsEnabled( )`
  - *autoProcessMouseWheel*  
`void autoProcessMouseWheel( java.awt.event.MouseWheelEvent )`
  - *autoTransferFocus*  
`final void autoTransferFocus( boolean )`

- *bounds*  
public Rectangle bounds( )
- *checkGD*  
void checkGD( java.lang.String )
- *checkImage*  
public int checkImage( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean checkWindowClosingException( )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *coalesceEvents*  
protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String constructComponentName( )
- *contains*  
public boolean contains( int , int )
- *contains*  
public boolean contains( java.awt.Point )
- *containsFocus*  
boolean containsFocus( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createBufferStrategy*  
void createBufferStrategy( int )
- *createBufferStrategy*  
void createBufferStrategy( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image createImage( java.awt.image.ImageProducer )
- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )

- *dispatchEvent*  
public final void **dispatchEvent**( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void **dispatchEventImpl**( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean **dispatchMouseWheelToAncestor**( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void **doAutoTransfer**( boolean    )
- *doLayout*  
public void **doLayout**( )
- *enable*  
public void **enable**( )
- *enable*  
public void **enable**( boolean    )
- *enableEvents*  
protected final void **enableEvents**( long    )
- *enableInputMethods*  
public void **enableInputMethods**( boolean    )
- *eventEnabled*  
boolean **eventEnabled**( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean **eventTypeEnabled**( int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , int    , int    )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String    , java.lang.Object    , java.lang.Object    )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getAccessibleIndexInParent*  
int **getAccessibleIndexInParent**( )
- *getAccessibleStateSet*  
AccessibleStateSet **getAccessibleStateSet**( )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle    )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )

- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener **getHierarchyBoundsListeners**( )
- *getHierarchyListeners*  
public synchronized HierarchyListener **getHierarchyListeners**( )
- *getIgnoreRepaint*  
public boolean **getIgnoreRepaint**( )
- *getInputContext*  
public InputContext **getInputContext**( )
- *getInputMethodListeners*  
public synchronized InputMethodListener **getInputMethodListeners**( )
- *getInputMethodRequests*  
public InputMethodRequests **getInputMethodRequests**( )
- *getKeyListeners*  
public synchronized KeyListener **getKeyListeners**( )

- *getListeners*  
public EventListener **getListeners**( java.lang.Class )
- *getLocale*  
public Locale **getLocale**( )
- *getLocation*  
public Point **getLocation**( )
- *getLocation*  
public Point **getLocation**( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point **getLocationOnScreen\_NoTreeLock**( )
- *getLocationOnScreen*  
public Point **getLocationOnScreen**( )
- *getMaximumSize*  
public Dimension **getMaximumSize**( )
- *getMinimumSize*  
public Dimension **getMinimumSize**( )
- *getMouseListeners*  
public synchronized MouseListener **getMouseListeners**( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener **getMouseMotionListeners**( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener **getMouseWheelListeners**( )
- *getName*  
public String **getName**( )
- *getNativeContainer*  
Container **getNativeContainer**( )
- *getParent\_NoClientCode*  
final Container **getParent\_NoClientCode**( )
- *getParent*  
public Container **getParent**( )
- *getPeer*  
public ComponentPeer **getPeer**( )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getSize*  
public Dimension **getSize**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getToolkitImpl*  
final Toolkit **getToolkitImpl**( )
- *getTreeLock*  
public final Object **getTreeLock**( )
- *getWidth*  
public int **getWidth**( )

- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *gotFocus*  
public boolean **gotFocus**( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean **handleEvent**( java.awt.Event )
- *hasFocus*  
public boolean **hasFocus**( )
- *hide*  
public void **hide**( )
- *imageUpdate*  
public boolean **imageUpdate**( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *inside*  
public boolean **inside**( int , int )
- *invalidate*  
public void **invalidate**( )
- *isBackgroundSet*  
public boolean **isBackgroundSet**( )
- *isCursorSet*  
public boolean **isCursorSet**( )
- *isDisplayable*  
public boolean **isDisplayable**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isEnabled*  
public boolean **isEnabled**( )
- *isEnabledImpl*  
final boolean **isEnabledImpl**( )
- *isFocusable*  
public boolean **isFocusable**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusOwner*  
public boolean **isFocusOwner**( )
- *isFocusTraversable*  
public boolean **isFocusTraversable**( )
- *isFocusTraversableOverridden*  
final boolean **isFocusTraversableOverridden**( )
- *isFontSet*  
public boolean **isFontSet**( )
- *isForegroundSet*  
public boolean **isForegroundSet**( )
- *isLightweight*  
public boolean **isLightweight**( )

- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )
- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )



- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvent*  
boolean postsOldMouseEvent( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )

- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *remove*  
public synchronized void remove( java.awt.MenuComponent    )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener    )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener    )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener    )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener    )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener    )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener    )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener    )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener    )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    , java.beans.PropertyChangeListener    )
- *repaint*  
public void repaint(    )
- *repaint*  
public void repaint( int    , int    , int    , int    )
- *repaint*  
public void repaint( long    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
protected boolean requestFocus( boolean    )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean    , boolean    )

- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )
- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )

- *setSize*  
public void setSize( java.awt.Dimension    )
- *setSize*  
public void setSize( int    , int    )
- *setVisible*  
public void setVisible( boolean    )
- *show*  
public void show(    )
- *show*  
public void show( boolean    )
- *size*  
public Dimension size(    )
- *toString*  
public String toString(    )
- *transferFocus*  
public void transferFocus(    )
- *transferFocusBackward*  
public void transferFocusBackward(    )
- *transferFocusUpCycle*  
public void transferFocusUpCycle(    )
- *update*  
public void update( java.awt.Graphics    )
- *updateCursorImmediately*  
final void updateCursorImmediately(    )
- *validate*  
public void validate(    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

### 13.2.45 CLASS URLChooser.BrowseButtonAction

---

The action associated with the 'browse' button

#### DECLARATION

---

```
protected class URLChooser.BrowseButtonAction
extends javax.swing.AbstractAction
```

#### CONSTRUCTORS

---

- *URLChooser.BrowseButtonAction*  
`protected URLChooser.BrowseButtonAction(    )`

#### METHODS

---

- *actionPerformed*  
`public void actionPerformed( java.awt.event.ActionEvent    e )`

METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object getValue( java.lang.String )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void **removePropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream )

**13.2.46 CLASS URLChooser.CancelButtonAction**

---

The action associated with the 'Cancel' button of the URL chooser

DECLARATION

---

```
protected class URLChooser.CancelButtonAction
extends javax.swing.AbstractAction
```

CONSTRUCTORS

---

- *URLChooser.CancelButtonAction*  
protected **URLChooser.CancelButtonAction**( )

METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
`public synchronized void addPropertyChangeListener(  
java.beans.PropertyChangeListener )`
- *clone*  
`protected Object clone( )`
- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String , java.lang.Object ,  
java.lang.Object )`
- *getKeys*  
`public Object getKeys( )`
- *getPropertyChangeListeners*  
`public synchronized PropertyChangeListener getPropertyChangeListeners( )`
- *getValue*  
`public Object getValue( java.lang.String )`
- *isEnabled*  
`public boolean isEnabled( )`
- *putValue*  
`public void putValue( java.lang.String , java.lang.Object )`
- *readObject*  
`private void readObject( java.io.ObjectInputStream )`
- *removePropertyChangeListener*  
`public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener )`
- *setEnabled*  
`public void setEnabled( boolean )`
- *writeObject*  
`private void writeObject( java.io.ObjectOutputStream )`

**13.2.47 CLASS URLChooser.ClearButtonAction**

---

The action associated with the 'Clear' button of the URL chooser

DECLARATION

---

```
protected class URLChooser.ClearButtonAction
extends javax.swing.AbstractAction
```

CONSTRUCTORS

---

- *URLChooser.ClearButtonAction*  
`protected URLChooser.ClearButtonAction( )`

METHODS

---

- *actionPerformed*  
`public void actionPerformed( java.awt.event.ActionEvent e )`

METHODS INHERITED FROM CLASS javax.swing.AbstractAction

---

- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *clone*  
protected Object clone( )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String   , java.lang.Object   ,   
java.lang.Object    )
- *getKeys*  
public Object getKeys( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getValue*  
public Object getValue( java.lang.String    )
- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String   , java.lang.Object    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *setEnabled*  
public void setEnabled( boolean    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

**13.2.48 CLASS URLChooser.DocumentAdapter**

---

To listen to the document changes

DECLARATION

---

```
protected class URLChooser.DocumentAdapter
extends java.lang.Object
implements javax.swing.event.DocumentListener
```

CONSTRUCTORS

---

- *URLChooser.DocumentAdapter*  
protected URLChooser.DocumentAdapter( )

## METHODS

- *changedUpdate*  
public void **changedUpdate**( javax.swing.event.DocumentEvent e )
- *insertUpdate*  
public void **insertUpdate**( javax.swing.event.DocumentEvent e )
- *removeUpdate*  
public void **removeUpdate**( javax.swing.event.DocumentEvent e )

## 13.2.49 CLASS URLChooser.OKButtonAction

The action associated with the 'OK' button of the URL chooser

## DECLARATION

```
protected class URLChooser.OKButtonAction
extends javax.swing.AbstractAction
```

## CONSTRUCTORS

- *URLChooser.OKButtonAction*  
protected **URLChooser.OKButtonAction**( )

## METHODS

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent e )

## METHODS INHERITED FROM CLASS javax.swing.AbstractAction

- *addPropertyChangeListener*  
public synchronized void **addPropertyChangeListener**(  
java.beans.PropertyChangeListener )
- *clone*  
protected Object **clone**( )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object ,  
java.lang.Object )
- *getKeys*  
public Object **getKeys**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getValue*  
public Object **getValue**( java.lang.String )



- *isEnabled*  
public boolean isEnabled( )
- *putValue*  
public void putValue( java.lang.String , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *setEnabled*  
public void setEnabled( boolean )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

## Chapter 14

# Package koala.dynamicjava.classfile

Package Contents

Page

---

### Classes

<b>AbstractMethodIdentifier</b> .....	1043
<i>The classes derived from this one are used to represents a method</i>	
<b>AttributeInfo</b> .....	1044
<i>The classes derived from this one represents bytecode attributes</i>	
<b>AttributeOwnerComponent</b> .....	1045
<i>This class represents a component of the bytecode 'ClassFile' format that contains attributes</i>	
<b>BytecodeComponent</b> .....	1046
<i>This class represents a component of the bytecode 'ClassFile' format</i>	
<b>ClassFile</b> .....	1047
<i>This class allows the creation of JVM bytecode class format outputs</i>	
<b>ClassIdentifier</b> .....	1050
<i>This class represents a class in the JVM internal format</i>	
<b>CodeAttribute</b> .....	1051
<i>This class represents a method code attribute</i>	
<b>CodeAttribute.ExceptionTableEntry</b> .....	1053
<i>...no description...</i>	
<b>ConstantPool</b> .....	1054
<i>This class represents JVM bytecode constant pools</i>	
<b>ConstantPool.ClassInfo</b> .....	1058
<i>This class is used to store class constants in the pool</i>	
<b>ConstantPool.DoubleInfo</b> .....	1059
<i>This class is used to store double constants in the pool</i>	
<b>ConstantPool.FieldInfo</b> .....	1060
<i>This class is used to store field constants in the pool</i>	
<b>ConstantPool.FloatInfo</b> .....	1061
<i>This class is used to store float constants in the pool</i>	
<b>ConstantPool.Info</b> .....	1062
<i>This class is used to store info in the pool</i>	
<b>ConstantPool.IntegerInfo</b> .....	1062
<i>This class is used to store integer constants in the pool</i>	
<b>ConstantPool.InterfaceMethodInfo</b> .....	1063
<i>This class is used to store interface method constants in the pool</i>	
<b>ConstantPool.LongInfo</b> .....	1064

<i>This class is used to store long constants in the pool</i>	
<b>ConstantPool.MethodInfo</b> .....	1065
<i>This class is used to store method constants in the pool</i>	
<b>ConstantPool.NameAndTypeInfo</b> .....	1066
<i>This class is used to store name and type constants in the pool</i>	
<b>ConstantPool.NameAndTypeKey</b> .....	1067
<i>This class is used as a key to store a name and type info in the map</i>	
<b>ConstantPool.StringInfo</b> .....	1067
<i>This class is used to store string constants in the pool</i>	
<b>ConstantPool.UTF8Info</b> .....	1068
<i>This class is used to store UTF8 constants in the pool</i>	
<b>ConstantString</b> .....	1069
<i>This class represents a constant string</i>	
<b>ConstantValueAttribute</b> .....	1070
<i>This class represents a constant field value</i>	
<b>ExceptionsAttribute</b> .....	1072
<i>This class represents a method exceptions attribute</i>	
<b>FieldIdentifier</b> .....	1074
<i>This class is used to represents a field</i>	
<b>FieldInfo</b> .....	1075
<i>This class allows the creation of JVM bytecode field format outputs</i>	
<b>InnerClassesAttribute</b> .....	1077
<i>This class represents an inner class attribute</i>	
<b>InnerClassesEntry</b> .....	1079
<i>This class represents an innerclasses entry in an innerclasses attribute</i>	
<b>InterfaceMethodIdentifier</b> .....	1080
<i>This class is used to represents an interface method</i>	
<b>JVMUtilities</b> .....	1082
<i>This interface contains utility functions.</i>	
<b>MemberIdentifier</b> .....	1083
<i>The classes derived from this one are used to represents class members</i>	
<b>MethodIdentifier</b> .....	1085
<i>This class is used to represents a method</i>	
<b>MethodInfo</b> .....	1086
<i>This class allows the creation of JVM bytecode method format outputs</i>	
<b>SimpleAttribute</b> .....	1089
<i>This class represents an attribute that is composed of his name</i>	

---

## 14.1 Classes

### 14.1.1 CLASS *AbstractMethodIdentifier*

---

The classes derived from this one are used to represents a method

#### DECLARATION

---

```
public abstract class AbstractMethodIdentifier
extends koala.dynamicjava.classfile.MemberIdentifier
```

#### FIELDS

---

- private String parameters
  - The parameters types

#### CONSTRUCTORS

---

- *AbstractMethodIdentifier*  

```
public AbstractMethodIdentifier( java.lang.String  dc, java.lang.String  n,
java.lang.String  t, java.lang.String [] p )
```

  - **Usage**
    - \* Creates a new method identifier
  - **Parameters**
    - \* **dc** - the declaring class of this member
    - \* **n** - the name of this member
    - \* **t** - the type of this member in JVM format
    - \* **p** - the parameters types

#### METHODS

---

- *equals*  

```
public boolean equals( java.lang.Object  other )
```

  - **Usage**
    - \* Indicates whether some other object is equal to this one
- *getParameters*  

```
public String getParameters( )
```

  - **Usage**
    - \* Returns the parameters types
- *hashCode*  

```
public int hashCode( )
```

  - **Usage**
    - \* Returns a hash code value for this object

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.MemberIdentifier`

---

( in 14.1.32, page 1083)

- *equals*  
`public boolean equals( java.lang.Object other )`  
 – **Usage**  
 \* Indicates whether some other object is equal to this one

---

- *getDeclaringClass*  
`public String getDeclaringClass( )`  
 – **Usage**  
 \* Returns the declaring class of this member

---

- *getName*  
`public String getName( )`  
 – **Usage**  
 \* Returns the name of this member

---

- *getType*  
`public String getType( )`  
 – **Usage**  
 \* Returns the type of this member in JVM format

---

- *hashCode*  
`public int hashCode( )`  
 – **Usage**  
 \* Returns a hash code value for this object

**14.1.2 CLASS AttributeInfo**

---

The classes derived from this one represents bytecode attributes

DECLARATION

---

```
public abstract class AttributeInfo
extends koala.dynamicjava.classfile.BytecodeComponent
```

CONSTRUCTORS

---

- *AttributeInfo*  
`public AttributeInfo( koala.dynamicjava.classfile.ConstantPool cp,  
 java.lang.String name )`  
 – **Usage**  
 \* Initializes the new attribute  
 – **Parameters**  
 \* **cp** - the constant pool where constants are stored  
 \* **name** - the name of this attribute

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.BytecodeComponent`

---

( in 14.1.4, page 1046)

- *getConstantPool*  
`public ConstantPool getConstantPool( )`  
  - **Usage**  
\* Returns the constant pool
- *write*  
`public abstract void write( java.io.DataOutputStream out )`  
  - **Usage**  
\* Writes the class file to the given output stream
- *write*  
`public void write( java.io.OutputStream out )`  
  - **Usage**  
\* Writes the class file to the given output stream

**14.1.3 CLASS AttributeOwnerComponent**

---

This class represents a component of the bytecode 'ClassFile' format that contains attributes

DECLARATION

---

```
public abstract class AttributeOwnerComponent
extends koala.dynamicjava.classfile.BytecodeComponent
```

CONSTRUCTORS

---

- *AttributeOwnerComponent*  
`protected AttributeOwnerComponent( )`  
  - **Usage**  
\* Initializes a new bytecode component
  - **Parameters**  
\* **cp** - the constant pool  
\* **af** - the access flags  
\* **ni** - the name index in the constant pool

METHODS

---

- *setAccessFlags*  
`public void setAccessFlags( int flags )`  
  - **Usage**  
\* Sets the access flags for this class

- 
- *setSyntheticAttribute*  
 public void **setSyntheticAttribute**( )  
 – **Usage**  
 \* Sets the synthetic attribute to this field

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.BytecodeComponent

---

( in 14.1.4, page 1046)

- *getConstantPool*  
 public ConstantPool **getConstantPool**( )  
 – **Usage**  
 \* Returns the constant pool
- *write*  
 public abstract void **write**( java.io.DataOutputStream out )  
 – **Usage**  
 \* Writes the class file to the given output stream
- *write*  
 public void **write**( java.io.OutputStream out )  
 – **Usage**  
 \* Writes the class file to the given output stream

### 14.1.4 CLASS BytecodeComponent

---

This class represents a component of the bytecode 'ClassFile' format

#### DECLARATION

---

<pre>public abstract class BytecodeComponent <b>extends</b> java.lang.Object</pre>
--

#### CONSTRUCTORS

---

- *BytecodeComponent*  
 protected **BytecodeComponent**( koala.dynamicjava.classfile.ConstantPool cp, short ni )  
 – **Usage**  
 \* Initializes a new bytecode component  
 – **Parameters**  
 \* cp - the constant pool  
 \* ni - the name index in the constant pool

METHODS

---

- *getConstantPool*  
`public ConstantPool getConstantPool( )`  
  - **Usage**  
\* Returns the constant pool

---
- *write*  
`public abstract void write( java.io.DataOutputStream out )`  
  - **Usage**  
\* Writes the class file to the given output stream

---
- *write*  
`public void write( java.io.OutputStream out )`  
  - **Usage**  
\* Writes the class file to the given output stream

**14.1.5 CLASS ClassFile**

---

This class allows the creation of JVM bytecode class format outputs

DECLARATION

---

```
public class ClassFile
extends koala.dynamicjava.classfile.AttributeOwnerComponent
```

FIELDS

---

- private static final int MAGIC  
  -
- private static final short MAJOR\_VERSION  
  -
- private static final short MINOR\_VERSION  
  -
- private short superClass  
  - The index of the name of the superclass in the constant pool
- private List interfaces  
  - The implemented interfaces
- private List fields



- The fields
- private List methods
  - The methods

## CONSTRUCTORS

---

- *ClassFile*  
public **ClassFile**( java.lang.String name, java.lang.String sname )
  - **Usage**
    - \* Creates a new ClassFile structure
  - **Parameters**
    - \* **name** - the name of this class
    - \* **sname** - the name of the superclass

## METHODS

---

- *addInterface*  
public void **addInterface**( java.lang.String name )
  - **Usage**
    - \* Adds an interface to the list of the implemented interfaces
- *createField*  
public **FieldInfo** **createField**( java.lang.String tp, java.lang.String nm )
  - **Usage**
    - \* Creates a new field
  - **Parameters**
    - \* **tp** - the type of the field
    - \* **nm** - the name of the field
  - **See Also**
    - \* koala.dynamicjava.classfile.FieldInfo.FieldInfo
- *createMethod*  
public **MethodInfo** **createMethod**( java.lang.String rt, java.lang.String nm, java.lang.String [] pt )
  - **Usage**
    - \* Creates a new method
  - **Parameters**
    - \* **rt** - the return type
    - \* **nm** - the name of the method
    - \* **pt** - the parameter types
  - **See Also**
    - \* koala.dynamicjava.classfile.MethodInfo.MethodInfo

- *getConstantPool*  
`public ConstantPool getConstantPool( )`
  - **Usage**
    - \* Returns the constant pool

---
- *setAbstract*  
`public void setAbstract( )`
  - **Usage**
    - \* Sets the abstract flag for this class

---
- *setFinal*  
`public void setFinal( )`
  - **Usage**
    - \* Sets the final flag for this class

---
- *setInnerClassesAttribute*  
`public void setInnerClassesAttribute(  
koala.dynamicjava.classfile.InnerClassesAttribute attr )`
  - **Usage**
    - \* Sets the innerclasses attribute to the class
  - **Parameters**
    - \* **attr** - the attribute to set

---
- *setInterface*  
`public void setInterface( )`
  - **Usage**
    - \* Sets the interface flag for this class

---
- *setPublic*  
`public void setPublic( )`
  - **Usage**
    - \* Sets the public flag for this class

---
- *setSuper*  
`public void setSuper( )`
  - **Usage**
    - \* Sets the super flag for this class

---
- *write*  
`public void write( java.io.DataOutputStream out )`
  - **Usage**
    - \* Writes the class file to the given output stream

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.AttributeOwnerComponent`

---

( in 14.1.3, page 1045)

- *setAccessFlags*  
 public void **setAccessFlags**( int flags )  
 – **Usage**  
   \* Sets the access flags for this class

---

- *setSyntheticAttribute*  
 public void **setSyntheticAttribute**( )  
 – **Usage**  
   \* Sets the synthetic attribute to this field

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.BytecodeComponent`

---

( in 14.1.4, page 1046)

- *getConstantPool*  
 public ConstantPool **getConstantPool**( )  
 – **Usage**  
   \* Returns the constant pool

---

- *write*  
 public abstract void **write**( java.io.DataOutputStream out )  
 – **Usage**  
   \* Writes the class file to the given output stream

---

- *write*  
 public void **write**( java.io.OutputStream out )  
 – **Usage**  
   \* Writes the class file to the given output stream

**14.1.6 CLASS `ClassIdentifier`**

---

This class represents a class in the JVM internal format

DECLARATION

---

```
public class ClassIdentifier
extends java.lang.Object
```

FIELDS

---

- private String value  
 – The value of this class identifier

CONSTRUCTORS

---

- *ClassIdentifier*  
`public ClassIdentifier( java.lang.String v )`
  - **Usage**  
 \* Creates a new class identifier

METHODS

---

- *equals*  
`public boolean equals( java.lang.Object other )`
  - **Usage**  
 \* Indicates whether some other object is equal to this one
- *getValue*  
`public String getValue( )`
  - **Usage**  
 \* Returns the value of this constant
- *hashCode*  
`public int hashCode( )`
  - **Usage**  
 \* Returns a hash code value for this object

**14.1.7 CLASS CodeAttribute**

---

This class represents a method code attribute

DECLARATION

---

```
public class CodeAttribute
extends koala.dynamicjava.classfile.AttributeInfo
```

FIELDS

---

- private short maxStack
  - The max depth of the operand stack
- private short maxLocals
  - The max number of local variables
- private byte code

- The code
- private List exceptionTable
  - The exception table
- private List attributes
  - The attributes

## CONSTRUCTORS

---

- *CodeAttribute*  
public **CodeAttribute**( koala.dynamicjava.classfile.ConstantPool cp )
  - **Usage**
    - \* Creates a new empty (not valid) code attribute
  - **Parameters**
    - \* cp - the constant pool

## METHODS

---

- *addExceptionTableEntry*  
public void **addExceptionTableEntry**( short spc, short epc, short tpc, java.lang.String ex )
  - **Usage**
    - \* Adds an exception entry in the exception table
  - **Parameters**
    - \* spc - the start of the try statement
    - \* epc - the end of the try statement
    - \* tpc - the handler position
    - \* ex - the name of the exception
- *setCode*  
public void **setCode**( byte [] code, short nl, short ms )
  - **Usage**
    - \* Sets the code for this code attribute
  - **Parameters**
    - \* code - the byte code array
    - \* nl - the number of local variables
    - \* ms - the max stack size
- *write*  
public void **write**( java.io.DataOutputStream out )
  - **Usage**
    - \* Writes the code info to the given output stream.

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.AttributeInfo`

---

( in 14.1.2, page 1044)

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.BytecodeComponent`

---

( in 14.1.4, page 1046)

- *getConstantPool*  
`public ConstantPool getConstantPool( )`  
  - **Usage**  
 \* Returns the constant pool
- *write*  
`public abstract void write( java.io.DataOutputStream out )`  
  - **Usage**  
 \* Writes the class file to the given output stream
- *write*  
`public void write( java.io.OutputStream out )`  
  - **Usage**  
 \* Writes the class file to the given output stream

**14.1.8 CLASS CodeAttribute.ExceptionTableEntry**

---

DECLARATION

---

```
class CodeAttribute.ExceptionTableEntry
extends java.lang.Object
```

FIELDS

---

- `private short startPc`  
  - The 'try' block starting position
- `private short endPc`  
  - The 'try' block end
- `private short handlerPc`  
  - The index of the 'catch' statement
- `private short catchType`  
  - The index of the name of the caught exception in the constant pool

---

CONSTRUCTORS

---

- *CodeAttribute.ExceptionTableEntry*

```
public CodeAttribute.ExceptionTableEntry( short  spc, short  epc, short  
hpc, short  ct )
```

- **Usage**

- \* Creates a new exception table entry

- **Parameters**

- \* **spc** - the 'try' block starting position

- \* **epc** - the 'try' block end

- \* **hpc** - the index of the 'catch' statement

- \* **ct** - the index of the name of the caught exception in the constant pool

---

METHODS

---

- *getLength*

```
public short getLength( )
```

- **Usage**

- \* Returns the length of the entry

---

- *write*

```
public void write( java.io.DataOutputStream  out )
```

- **Usage**

- \* Writes the field info to the given output stream.

---

- *write*

```
public void write( java.io.OutputStream  out )
```

- **Usage**

- \* Writes the field info to the given output stream

### 14.1.9 CLASS ConstantPool

---

This class represents JVM bytecode constant pools

---

DECLARATION

---

<pre>public class ConstantPool <b>extends</b> java.lang.Object</pre>
--

FIELDS

---

- private static final byte CONSTANT\_UTF8
  -
- private static final byte CONSTANT\_INTEGER
  -
- private static final byte CONSTANT\_FLOAT
  -
- private static final byte CONSTANT\_LONG
  -
- private static final byte CONSTANT\_DOUBLE
  -
- private static final byte CONSTANT\_CLASS
  -
- private static final byte CONSTANT\_STRING
  -
- private static final byte CONSTANT\_FIELDREF
  -
- private static final byte CONSTANT\_METHODREF
  -
- private static final byte CONSTANT\_INTERFACEMETHODREF
  -
- private static final byte CONSTANT\_NAMEANDTYPE
  -
- private Map constants
  - The constants
- private short count
  - The constant count

CONSTRUCTORS

---

- *ConstantPool*  
**public ConstantPool( )**
  - **Usage**
    - \* Creates a new constant pool



METHODS

---

- *get*  
`ConstantPool.Info get( short i )`  
  - **Usage**  
 \* Returns the entry at the given position in the pool

---
- *getCount*  
`public short getCount( )`  
  - **Usage**  
 \* Returns the constant pool count according to the JVM Spec.

---
- *put*  
`public short put( koala.dynamicjava.classfile.ClassIdentifier cst )`  
  - **Usage**  
 \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.
  - **Parameters**  
 \* `cst` - the constant to add
  - **Returns** - the index of the constant in the pool

---
- *put*  
`public short put( koala.dynamicjava.classfile.ConstantString cst )`  
  - **Usage**  
 \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.
  - **Parameters**  
 \* `cst` - the constant to add
  - **Returns** - the index of the constant in the pool

---
- *put*  
`public short put( java.lang.Double cst )`  
  - **Usage**  
 \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.
  - **Parameters**  
 \* `cst` - the constant to add
  - **Returns** - the index of the constant in the pool

---
- *put*  
`public short put( koala.dynamicjava.classfile.FieldIdentifier cst )`  
  - **Usage**  
 \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.

- **Parameters**

- \* **cst** - the constant to add

- **Returns** - the index of the constant in the pool

---

- *put*

```
public short put( java.lang.Float  cst )
```

- **Usage**

- \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.

- **Parameters**

- \* **cst** - the constant to add

- **Returns** - the index of the constant in the pool

---

- *put*

```
public short put( java.lang.Integer  cst )
```

- **Usage**

- \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.

- **Parameters**

- \* **cst** - the constant to add

- **Returns** - the index of the constant in the pool

---

- *put*

```
public short put( koala.dynamicjava.classfile.InterfaceMethodIdentifier  cst )
```

- **Usage**

- \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.

- **Parameters**

- \* **cst** - the constant to add

- **Returns** - the index of the constant in the pool

---

- *put*

```
public short put( java.lang.Long  cst )
```

- **Usage**

- \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.

- **Parameters**

- \* **cst** - the constant to add

- **Returns** - the index of the constant in the pool

---

- *put*

```
public short put( koala.dynamicjava.classfile.MethodIdentifier  cst )
```

- **Usage**

- \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.
  - **Parameters**
    - \* **cst** - the constant to add
  - **Returns** - the index of the constant in the pool

---
- *putNameAndType*

```
short putNameAndType( java.lang.String name, java.lang.String type,
java.lang.String [] params )
```

  - **Usage**
    - \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.
  - **Parameters**
    - \* **name** - the name
    - \* **type** - the type
    - \* **type** - the type of the parameters
  - **Returns** - the index of the constant in the pool

---
- *putUTF8*

```
short putUTF8( java.lang.String cst )
```

  - **Usage**
    - \* Adds a constant to the pool. If the constant is already present in the pool, do nothing.
  - **Parameters**
    - \* **cst** - the constant to add
  - **Returns** - the index of the constant in the pool

---
- *write*

```
public void write( java.io.DataOutputStream out )
```

  - **Usage**
    - \* Writes the content of this pool to the given output stream

---
- *write*

```
public void write( java.io.OutputStream out )
```

  - **Usage**
    - \* Writes the content of this pool to the given output stream

#### 14.1.10 CLASS ConstantPool.ClassInfo

---

This class is used to store class constants in the pool

##### DECLARATION

---

```
class ConstantPool.ClassInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.ClassInfo*  
**ConstantPool.ClassInfo**( koala.dynamicjava.classfile.ClassIdentifier v )  
 – **Usage**  
 \* Creates a new class constant information object

METHODS

---

- *write*  
**void write**( java.io.DataOutputStream out )  
 – **Usage**  
 \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.ConstantPool.Info

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
**short getIndexIncrement**( )  
 – **Usage**  
 \* Returns the index increment for this type of info
- 
- *write*  
**abstract void write**( java.io.DataOutputStream out )  
 – **Usage**  
 \* Writes the constant information to the specified stream

**14.1.11 CLASS ConstantPool.DoubleInfo**

---

This class is used to store double constants in the pool

DECLARATION

---

```
class ConstantPool.DoubleInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.DoubleInfo*  
**ConstantPool.DoubleInfo**( java.lang.Double v )  
 – **Usage**  
 \* Creates a new double constant information object

METHODS

---

- *getIndexIncrement*  
**short getIndexIncrement( )**
    - **Usage**
      - \* Returns the index increment for this type of info
- 
- *write*  
**void write( java.io.DataOutputStream out )**
    - **Usage**
      - \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.ConstantPool.Info

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
**short getIndexIncrement( )**
    - **Usage**
      - \* Returns the index increment for this type of info
- 
- *write*  
**abstract void write( java.io.DataOutputStream out )**
    - **Usage**
      - \* Writes the constant information to the specified stream

**14.1.12 CLASS ConstantPool.FieldInfo**

---

This class is used to store field constants in the pool

DECLARATION

---

```
class ConstantPool.FieldInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.FieldInfo*  
**ConstantPool.FieldInfo( koala.dynamicjava.classfile.FieldIdentifier v )**
  - **Usage**
    - \* Creates a new field constant information object

METHODS

---

- *write*  
**void write( java.io.DataOutputStream out )**
  - **Usage**
    - \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.ConstantPool.Info`

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
**short getIndexIncrement( )**
    - **Usage**
      - \* Returns the index increment for this type of info
- 
- *write*  
**abstract void write( java.io.DataOutputStream out )**
    - **Usage**
      - \* Writes the constant information to the specified stream

**14.1.13 CLASS ConstantPool.FloatInfo**

---

This class is used to store float constants in the pool

DECLARATION

---

```
class ConstantPool.FloatInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.FloatInfo*  
**ConstantPool.FloatInfo( java.lang.Float v )**
  - **Usage**
    - \* Creates a new float constant information object

METHODS

---

- *write*  
**void write( java.io.DataOutputStream out )**
  - **Usage**
    - \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.ConstantPool.Info`

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
`short getIndexIncrement( )`  
  - **Usage**  
 \* Returns the index increment for this type of info
- *write*  
`abstract void write( java.io.DataOutputStream out )`  
  - **Usage**  
 \* Writes the constant information to the specified stream

**14.1.14 CLASS ConstantPool.Info**

---

This class is used to store info in the pool

DECLARATION

---

```
abstract class ConstantPool.Info
extends java.lang.Object
```

CONSTRUCTORS

---

- *ConstantPool.Info*  
`ConstantPool.Info( java.lang.Object cst )`  
  - **Usage**  
 \* Initializes the object

METHODS

---

- *getIndexIncrement*  
`short getIndexIncrement( )`  
  - **Usage**  
 \* Returns the index increment for this type of info
- *write*  
`abstract void write( java.io.DataOutputStream out )`  
  - **Usage**  
 \* Writes the constant information to the specified stream

**14.1.15 CLASS ConstantPool.IntegerInfo**

---

This class is used to store integer constants in the pool

DECLARATION

---

```
class ConstantPool.IntegerInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.IntegerInfo*  
**ConstantPool.IntegerInfo**( java.lang.Integer v )
  - **Usage**
    - \* Creates a new integer constant information object

METHODS

---

- *write*  
**void write**( java.io.DataOutputStream out )
  - **Usage**
    - \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.ConstantPool.Info

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
**short getIndexIncrement**( )
  - **Usage**
    - \* Returns the index increment for this type of info
- *write*  
**abstract void write**( java.io.DataOutputStream out )
  - **Usage**
    - \* Writes the constant information to the specified stream

**14.1.16 CLASS ConstantPool.InterfaceMethodInfo**

---

This class is used to store interface method constants in the pool

DECLARATION

---

```
class ConstantPool.InterfaceMethodInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```



CONSTRUCTORS

---

- *ConstantPool.InterfaceMethodInfo*

**ConstantPool.InterfaceMethodInfo**(  
koala.dynamicjava.classfile.InterfaceMethodIdentifier v )

– **Usage**

- \* Creates a new method constant information object

METHODS

---

- *write*

**void write**( java.io.DataOutputStream out )

– **Usage**

- \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.ConstantPool.Info

---

( in 14.1.14, page 1062)

- *getIndexIncrement*

**short getIndexIncrement**( )

– **Usage**

- \* Returns the index increment for this type of info

- *write*

**abstract void write**( java.io.DataOutputStream out )

– **Usage**

- \* Writes the constant information to the specified stream

**14.1.17 CLASS ConstantPool.LongInfo**

---

This class is used to store long constants in the pool

DECLARATION

---

```
class ConstantPool.LongInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.LongInfo*

**ConstantPool.LongInfo**( java.lang.Long v )

– **Usage**

- \* Creates a new long constant information object

METHODS

---

- *getIndexIncrement*  
**short getIndexIncrement( )**
    - **Usage**
      - \* Returns the index increment for this type of info
- 
- *write*  
**void write( java.io.DataOutputStream out )**
    - **Usage**
      - \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.ConstantPool.Info

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
**short getIndexIncrement( )**
    - **Usage**
      - \* Returns the index increment for this type of info
- 
- *write*  
**abstract void write( java.io.DataOutputStream out )**
    - **Usage**
      - \* Writes the constant information to the specified stream

**14.1.18 CLASS ConstantPool.MethodInfo**

---

This class is used to store method constants in the pool

DECLARATION

---

```
class ConstantPool.MethodInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.MethodInfo*  
**ConstantPool.MethodInfo( koala.dynamicjava.classfile.MethodIdentifier v )**
  - **Usage**
    - \* Creates a new method constant information object

METHODS

---

- *write*  
`void write( java.io.DataOutputStream out )`  
 – **Usage**  
 \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.ConstantPool.Info`

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
`short getIndexIncrement( )`  
 – **Usage**  
 \* Returns the index increment for this type of info
- 
- *write*  
`abstract void write( java.io.DataOutputStream out )`  
 – **Usage**  
 \* Writes the constant information to the specified stream

**14.1.19 CLASS ConstantPool.NameAndTypeInfo**

---

This class is used to store name and type constants in the pool

DECLARATION

---

```
class ConstantPool.NameAndTypeInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.NameAndTypeInfo*  
`ConstantPool.NameAndTypeInfo( koala.dynamicjava.classfile.ConstantPool.NameAndTypeKey v )`  
 – **Usage**  
 \* Creates a new class constant information object

METHODS

---

- *write*  
`void write( java.io.DataOutputStream out )`  
 – **Usage**  
 \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.ConstantPool.Info`

( in 14.1.14, page 1062)

- *getIndexIncrement*  
`short getIndexIncrement( )`
  - **Usage**  
 \* Returns the index increment for this type of info
- *write*  
`abstract void write( java.io.DataOutputStream out )`
  - **Usage**  
 \* Writes the constant information to the specified stream

**14.1.20 CLASS ConstantPool.NameAndTypeKey**

This class is used as a key to store a name and type info in the map

## DECLARATION

```
static class ConstantPool.NameAndTypeKey
extends java.lang.Object
```

## CONSTRUCTORS

- *ConstantPool.NameAndTypeKey*  
`ConstantPool.NameAndTypeKey( java.lang.String n, java.lang.String t,  
 java.lang.String [] p )`

## METHODS

- *equals*  
`public boolean equals( java.lang.Object other )`
- *hashCode*  
`public int hashCode( )`

**14.1.21 CLASS ConstantPool.StringInfo**

This class is used to store string constants in the pool

## DECLARATION

```
class ConstantPool.StringInfo
extends koala.dynamicjava.classfile.ConstantPool.Info
```

CONSTRUCTORS

---

- *ConstantPool.StringInfo*  
**ConstantPool.StringInfo**( koala.dynamicjava.classfile.ConstantString v )
  - **Usage**
    - \* Creates a new string constant information object

METHODS

---

- *write*  
**void write**( java.io.DataOutputStream out )
  - **Usage**
    - \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.ConstantPool.Info

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
**short getIndexIncrement**( )
  - **Usage**
    - \* Returns the index increment for this type of info
- *write*  
**abstract void write**( java.io.DataOutputStream out )
  - **Usage**
    - \* Writes the constant information to the specified stream

**14.1.22 CLASS ConstantPool.UTF8Info**

---

This class is used to store UTF8 constants in the pool

DECLARATION

---

<pre>class ConstantPool.UTF8Info <b>extends</b> koala.dynamicjava.classfile.ConstantPool.Info</pre>
---

CONSTRUCTORS

---

- *ConstantPool.UTF8Info*  
**ConstantPool.UTF8Info**( java.lang.String v )
  - **Usage**
    - \* Creates a new string constant information object

METHODS

---

- *write*  
 void **write**( java.io.DataOutputStream out )  
 – **Usage**  
 \* Writes the constant information to the specified stream

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.ConstantPool.Info

---

( in 14.1.14, page 1062)

- *getIndexIncrement*  
 short **getIndexIncrement**( )  
 – **Usage**  
 \* Returns the index increment for this type of info
- 
- *write*  
 abstract void **write**( java.io.DataOutputStream out )  
 – **Usage**  
 \* Writes the constant information to the specified stream

**14.1.23 CLASS ConstantString**

---

This class represents a constant string

DECLARATION

---

```
public class ConstantString
extends java.lang.Object
```

FIELDS

---

- private String value  
 – The value of this constant

CONSTRUCTORS

---

- *ConstantString*  
 public **ConstantString**( java.lang.String v )  
 – **Usage**  
 \* Creates a new constant

## METHODS

- *equals*  
`public boolean equals( java.lang.Object other )`  
 – **Usage**  
 \* Indicates whether some other object is equal to this one

---

- *getValue*  
`public String getValue( )`  
 – **Usage**  
 \* Returns the value of this constant

---

- *hashCode*  
`public int hashCode( )`  
 – **Usage**  
 \* Returns a hash code value for this object

## 14.1.24 CLASS ConstantValueAttribute

This class represents a constant field value

## DECLARATION

```
public class ConstantValueAttribute
extends koala.dynamicjava.classfile.AttributeInfo
```

## FIELDS

- private short index  
 – The index of this constant in the constant pool

## CONSTRUCTORS

- *ConstantValueAttribute*  
`private ConstantValueAttribute( koala.dynamicjava.classfile.ConstantPool cp )`  
 – **Usage**  
 \* Initializes a new constant value attribute  
 – **Parameters**  
 \* `cp` - the constant pool where constants are stored

- *ConstantValueAttribute*

```
public ConstantValueAttribute( koala.dynamicjava.classfile.ConstantPool
cp, java.lang.Double value )
```

- **Usage**

- \* Creates a new constant value attribute

- **Parameters**

- \* **cp** - the constant pool where constants are stored
    - \* **value** - the value of this constant

---

- *ConstantValueAttribute*

```
public ConstantValueAttribute( koala.dynamicjava.classfile.ConstantPool
cp, java.lang.Float value )
```

- **Usage**

- \* Creates a new constant value attribute

- **Parameters**

- \* **cp** - the constant pool where constants are stored
    - \* **value** - the value of this constant

---

- *ConstantValueAttribute*

```
public ConstantValueAttribute( koala.dynamicjava.classfile.ConstantPool
cp, java.lang.Integer value )
```

- **Usage**

- \* Creates a new constant value attribute

- **Parameters**

- \* **cp** - the constant pool where constants are stored
    - \* **value** - the value of this constant

---

- *ConstantValueAttribute*

```
public ConstantValueAttribute( koala.dynamicjava.classfile.ConstantPool
cp, java.lang.Long value )
```

- **Usage**

- \* Creates a new constant value attribute

- **Parameters**

- \* **cp** - the constant pool where constants are stored
    - \* **value** - the value of this constant

---

- *ConstantValueAttribute*

```
public ConstantValueAttribute( koala.dynamicjava.classfile.ConstantPool
cp, java.lang.String value )
```

- **Usage**

- \* Creates a new constant value attribute

- **Parameters**

- \* **cp** - the constant pool where constants are stored
    - \* **value** - the value of this constant



METHODS

---

- *write*  
`public void write( java.io.DataOutputStream out )`  
 – **Usage**  
 \* Writes the constant info to the given output stream.

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.AttributeInfo`

---

( in 14.1.2, page 1044)

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.BytecodeComponent`

---

( in 14.1.4, page 1046)

- *getConstantPool*  
`public ConstantPool getConstantPool( )`  
 – **Usage**  
 \* Returns the constant pool

---

- *write*  
`public abstract void write( java.io.DataOutputStream out )`  
 – **Usage**  
 \* Writes the class file to the given output stream

---

- *write*  
`public void write( java.io.OutputStream out )`  
 – **Usage**  
 \* Writes the class file to the given output stream

**14.1.25 CLASS ExceptionsAttribute**

---

This class represents a method exceptions attribute

DECLARATION

---

```
public class ExceptionsAttribute
extends koala.dynamicjava.classfile.AttributeInfo
```

FIELDS

---

- private List exceptionIndexTable  
 – The exception index table

CONSTRUCTORS

---

- *ExceptionsAttribute*

**public ExceptionsAttribute( koala.dynamicjava.classfile.ConstantPool cp )**

- **Usage**

- \* Creates a new empty (not valid) code attribute

- **Parameters**

- \* cp - the constant pool

METHODS

---

- *addException*

**public void addException( java.lang.String name )**

- **Usage**

- \* Adds an exception to the attribute

- **Parameters**

- \* name - the name of the exception

---

- *write*

**public void write( java.io.DataOutputStream out )**

- **Usage**

- \* Writes the exception info to the given output stream.

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.AttributeInfo

---

( in 14.1.2, page 1044)

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.BytecodeComponent

---

( in 14.1.4, page 1046)

- *getConstantPool*

**public ConstantPool getConstantPool( )**

- **Usage**

- \* Returns the constant pool

---

- *write*

**public abstract void write( java.io.DataOutputStream out )**

- **Usage**

- \* Writes the class file to the given output stream

---

- *write*

**public void write( java.io.OutputStream out )**

- **Usage**

- \* Writes the class file to the given output stream

### 14.1.26 CLASS FieldIdentifier

---

This class is used to represents a field

#### DECLARATION

---

```
public class FieldIdentifier
extends koala.dynamicjava.classfile.MemberIdentifier
```

#### CONSTRUCTORS

---

- *FieldIdentifier*  

```
public FieldIdentifier( java.lang.String dc, java.lang.String n,
java.lang.String t )
```

  - **Usage**
    - \* Creates a new field identifier
  - **Parameters**
    - \* **dc** - the declaring class of this member
    - \* **n** - the name of this member
    - \* **t** - the type of this member in JVM format

#### METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.MemberIdentifier

---

( in 14.1.32, page 1083)

- *equals*  

```
public boolean equals( java.lang.Object other )
```

  - **Usage**
    - \* Indicates whether some other object is equal to this one
- *getDeclaringClass*  

```
public String getDeclaringClass( )
```

  - **Usage**
    - \* Returns the declaring class of this member
- *getName*  

```
public String getName( )
```

  - **Usage**
    - \* Returns the name of this member
- *getType*  

```
public String getType( )
```

  - **Usage**
    - \* Returns the type of this member in JVM format
- *hashCode*  

```
public int hashCode( )
```

  - **Usage**
    - \* Returns a hash code value for this object

### 14.1.27 CLASS FieldInfo

---

This class allows the creation of JVM bytecode field format outputs

#### DECLARATION

---

```
public class FieldInfo
extends koala.dynamicjava.classfile.AttributeOwnerComponent
```

#### FIELDS

---

- private short descriptorIndex
  - The descriptor index

#### CONSTRUCTORS

---

- *FieldInfo*

```
public FieldInfo( koala.dynamicjava.classfile.ConstantPool  cp,
                  java.lang.String  tp, java.lang.String  nm )
```

  - **Usage**
    - \* Creates a new field info
  - **Parameters**
    - \* **cp** - the constant pool where constants are stored
    - \* **tp** - the type name. The type name must be fully qualified.
    - \* **nm** - the name of the field

#### METHODS

---

- *setConstantValueAttribute*

```
public void setConstantValueAttribute( java.lang.Double  value )
```

  - **Usage**
    - \* Sets the constant value attribute for this field to a double value.
- *setConstantValueAttribute*

```
public void setConstantValueAttribute( java.lang.Float  value )
```

  - **Usage**
    - \* Sets the constant value attribute for this field to a float value.
- *setConstantValueAttribute*

```
public void setConstantValueAttribute( java.lang.Integer  value )
```

---

– **Usage**

\* Sets the constant value attribute for this field to an integer value.

---

• *setConstantValueAttribute*

**public void setConstantValueAttribute( java.lang.Long value )**

– **Usage**

\* Sets the constant value attribute for this field to a long value.

---

• *setConstantValueAttribute*

**public void setConstantValueAttribute( java.lang.String value )**

– **Usage**

\* Sets the constant value attribute for this field to a string value.

---

• *setFinal*

**public void setFinal( )**

– **Usage**

\* Sets the final flag for this class

---

• *setPrivate*

**public void setPrivate( )**

– **Usage**

\* Sets the private flag for this class

---

• *setProtected*

**public void setProtected( )**

– **Usage**

\* Sets the protected flag for this class

---

• *setPublic*

**public void setPublic( )**

– **Usage**

\* Sets the public flag for this class

---

• *setStatic*

**public void setStatic( )**

– **Usage**

\* Sets the static flag for this class

---

• *setTransient*

**public void setTransient( )**

– **Usage**

\* Sets the transient flag for this class

---

• *setVolatile*

**public void setVolatile( )**

---

– **Usage**

\* Sets the volatile flag for this class

---

• *write*

**public void write( java.io.DataOutputStream out )**

– **Usage**

\* Writes the field info to the given output stream

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.AttributeOwnerComponent`

---

( in 14.1.3, page 1045)

• *setAccessFlags*

**public void setAccessFlags( int flags )**

– **Usage**

\* Sets the access flags for this class

---

• *setSyntheticAttribute*

**public void setSyntheticAttribute( )**

– **Usage**

\* Sets the synthetic attribute to this field

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.BytecodeComponent`

---

( in 14.1.4, page 1046)

• *getConstantPool*

**public ConstantPool getConstantPool( )**

– **Usage**

\* Returns the constant pool

---

• *write*

**public abstract void write( java.io.DataOutputStream out )**

– **Usage**

\* Writes the class file to the given output stream

---

• *write*

**public void write( java.io.OutputStream out )**

– **Usage**

\* Writes the class file to the given output stream

## 14.1.28 CLASS InnerClassesAttribute

---

This class represents an inner class attribute

### DECLARATION

---

```
public class InnerClassesAttribute
extends koala.dynamicjava.classfile.AttributeInfo
```

FIELDS

---

- private List classes
  - The classes

CONSTRUCTORS

---

- *InnerClassesAttribute*  
 public **InnerClassesAttribute**( koala.dynamicjava.classfile.ConstantPool cp )  
  - **Usage**
    - \* Creates a new innerclasses attribute
  - **Parameters**
    - \* cp - the constant pool

METHODS

---

- *addInnerClassesEntry*  
 public InnerClassesEntry **addInnerClassesEntry**( )  
    - **Usage**
      - \* Adds an innerclasses entry to this attribute
- 
- *write*  
 public void **write**( java.io.DataOutputStream out )  
    - **Usage**
      - \* Writes this attribute to the given output stream.

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.AttributeInfo

---

( in 14.1.2, page 1044)

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.BytecodeComponent

---

( in 14.1.4, page 1046)

- *getConstantPool*  
 public ConstantPool **getConstantPool**( )  
    - **Usage**
      - \* Returns the constant pool
- 
- *write*  
 public abstract void **write**( java.io.DataOutputStream out )  
    - **Usage**
      - \* Writes the class file to the given output stream

- 
- *write*  
 public void **write**( java.io.OutputStream out )
    - **Usage**
      - \* Writes the class file to the given output stream

### 14.1.29 CLASS InnerClassesEntry

---

This class represents an innerclasses entry in an innerclasses attribute

#### DECLARATION

---

```
public class InnerClassesEntry
extends java.lang.Object
```

#### FIELDS

---

- private short innerClassInfoIndex
  - The inner class info index
- private short outerClassInfoIndex
  - The outer class info index
- private short innerNameIndex
  - The inner name index
- private short innerClassAccessFlags
  - The inner class access flags
- private ConstantPool constantPool
  - The constant pool used to store the constants

#### CONSTRUCTORS

---

- *InnerClassesEntry*  
 public **InnerClassesEntry**( koala.dynamicjava.classfile.ConstantPool cp )
  - **Usage**
    - \* Creates a new entry



## METHODS

- 
- *setInnerClassAccessFlags*  
 public void **setInnerClassAccessFlags**( short af )  
 – **Parameters**  
 \* af - the access flags

---

  - *setInnerClassInfo*  
 public void **setInnerClassInfo**( java.lang.String cname )  
 – **Usage**  
 \* Sets the inner class info  
 – **Parameters**  
 \* cname - the inner class name

---

  - *setInnerName*  
 public void **setInnerName**( java.lang.String name )  
 – **Usage**  
 \* Sets the inner class name  
 – **Parameters**  
 \* the - name of the inner class

---

  - *setOuterClassInfo*  
 public void **setOuterClassInfo**( java.lang.String cname )  
 – **Usage**  
 \* Sets the outer class info  
 – **Parameters**  
 \* cname - the outer class name

---

  - *write*  
 public void **write**( java.io.DataOutputStream out )  
 – **Usage**  
 \* Writes the code represented by this object to the given output stream.

## 14.1.30 CLASS InterfaceMethodIdentifier

---

This class is used to represents an interface method

## DECLARATION

---

```
public class InterfaceMethodIdentifier
extends koala.dynamicjava.classfile.AbstractMethodIdentifier
```

CONSTRUCTORS

---

• *InterfaceMethodIdentifier*

```
public InterfaceMethodIdentifier( java.lang.String  dc, java.lang.String  n,
    java.lang.String  t, java.lang.String [] p )
```

– **Usage**

\* Creates a new interface method identifier

– **Parameters**

\* **dc** - the declaring class of this member

\* **n** - the name of this member

\* **t** - the type of this member in JVM format

\* **p** - the parameters types

## METHODS INHERITED FROM CLASS

```
koala.dynamicjava.classfile.AbstractMethodIdentifier
```

---

( in 14.1.1, page 1043)

• *equals*

```
public boolean equals( java.lang.Object  other )
```

– **Usage**

\* Indicates whether some other object is equal to this one

• *getParameters*

```
public String getParameters( )
```

– **Usage**

\* Returns the parameters types

• *hashCode*

```
public int hashCode( )
```

– **Usage**

\* Returns a hash code value for this object

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.MemberIdentifier

---

( in 14.1.32, page 1083)

• *equals*

```
public boolean equals( java.lang.Object  other )
```

– **Usage**

\* Indicates whether some other object is equal to this one

• *getDeclaringClass*

```
public String getDeclaringClass( )
```

– **Usage**

\* Returns the declaring class of this member

• *getName*

```
public String getName( )
```

– **Usage**

- \* Returns the name of this member
- - *getType*  
`public String getType( )`
    - **Usage**
      - \* Returns the type of this member in JVM format
- - *hashCode*  
`public int hashCode( )`
    - **Usage**
      - \* Returns a hash code value for this object

### 14.1.31 CLASS JVMUtilities

---

This interface contains utility functions.

#### DECLARATION

---

```
public abstract class JVMUtilities
extends java.lang.Object
```

#### FIELDS

---

- private static final Map types
  - The table of the base type characters. Keys are classes
- private static final Map stypes
  - The table of the base type characters. Keys are strings

#### CONSTRUCTORS

---

- *JVMUtilities*  
`private JVMUtilities( )`
  - **Usage**
    - \* No need to create instances of this class

#### METHODS

---

- *createMethodDescriptor*  
`public static String createMethodDescriptor( java.lang.String rt,  
 java.lang.String [] pt )`
  - **Usage**
    - \* Creates a method descriptor

---

– **Parameters**

- \* **rt** - the return type name as returned by `getReturnTypeName`
  - \* **pt** - the parameters type names as returned by `getParameterTypeName`
- 

- *getName*

```
public static String getName( java.lang.Class c )
```

– **Usage**

- \* Returns the string that represents internally the given class
- 

- *getName*

```
public static String getName( java.lang.String c )
```

– **Usage**

- \* Returns the string that represents internally the given class name
- 

- *getParameterTypeName*

```
public static String getParameterTypeName( java.lang.Class c )
```

– **Usage**

- \* Returns the string that represents internally the given class
- 

- *getParameterTypeName*

```
public static String getParameterTypeName( java.lang.String c )
```

– **Usage**

- \* Returns the string that represents internally the given class name
- 

- *getReturnTypeName*

```
public static String getReturnTypeName( java.lang.Class c )
```

– **Usage**

- \* Returns the string that represents internally the given class
- 

- *getReturnTypeName*

```
public static String getReturnTypeName( java.lang.String c )
```

– **Usage**

- \* Returns the string that represents internally the given class name

### 14.1.32 CLASS MemberIdentifier

---

The classes derived from this one are used to represents class members

#### DECLARATION

---

```
public abstract class MemberIdentifier
extends java.lang.Object
```

## FIELDS

---

- private String declaringClass
  - The declaring class for this member
- private String name
  - The name of this member
- private String type
  - The type of this member in JVM format

## CONSTRUCTORS

---

- *MemberIdentifier*  
`public MemberIdentifier( java.lang.String dc, java.lang.String n,  
java.lang.String t )`
  - **Usage**
    - \* Initializes the identifier
  - **Parameters**
    - \* **dc** - the declaring class of this member
    - \* **n** - the name of this member
    - \* **t** - the type of this member in JVM format

## METHODS

---

- *equals*  
`public boolean equals( java.lang.Object other )`
  - **Usage**
    - \* Indicates whether some other object is equal to this one
- *getDeclaringClass*  
`public String getDeclaringClass( )`
  - **Usage**
    - \* Returns the declaring class of this member
- *getName*  
`public String getName( )`
  - **Usage**
    - \* Returns the name of this member
- *getType*  
`public String getType( )`
  - **Usage**

\* Returns the type of this member in JVM format

---

- *hashCode*

**public int hashCode( )**

- **Usage**

\* Returns a hash code value for this object

### 14.1.33 CLASS MethodIdentifier

---

This class is used to represents a method

#### DECLARATION

---

```
public class MethodIdentifier
extends koala.dynamicjava.classfile.AbstractMethodIdentifier
```

#### CONSTRUCTORS

---

- *MethodIdentifier*

**public MethodIdentifier( java.lang.String dc, java.lang.String n,  
java.lang.String t, java.lang.String [] p )**

- **Usage**

\* Creates a new method identifier

- **Parameters**

\* **dc** - the declaring class of this member  
 \* **n** - the name of this member  
 \* **t** - the return type of this member in JVM format  
 \* **p** - the parameters types

#### METHODS INHERITED FROM CLASS

koala.dynamicjava.classfile.AbstractMethodIdentifier

---

( in 14.1.1, page 1043)

- *equals*

**public boolean equals( java.lang.Object other )**

- **Usage**

\* Indicates whether some other object is equal to this one

---

- *getParameters*

**public String getParameters( )**

- **Usage**

\* Returns the parameters types

---

- *hashCode*

**public int hashCode( )**

- **Usage**

\* Returns a hash code value for this object

METHODS INHERITED FROM CLASS `koala.dynamicjava.classfile.MemberIdentifier`

---

( in 14.1.32, page 1083)

- *equals*  
`public boolean equals( java.lang.Object other )`  
  - **Usage**  
\* Indicates whether some other object is equal to this one
- *getDeclaringClass*  
`public String getDeclaringClass( )`  
  - **Usage**  
\* Returns the declaring class of this member
- *getName*  
`public String getName( )`  
  - **Usage**  
\* Returns the name of this member
- *getType*  
`public String getType( )`  
  - **Usage**  
\* Returns the type of this member in JVM format
- *hashCode*  
`public int hashCode( )`  
  - **Usage**  
\* Returns a hash code value for this object

### 14.1.34 CLASS MethodInfo

---

This class allows the creation of JVM bytecode method format outputs

#### DECLARATION

---

```
public class MethodInfo
extends koala.dynamicjava.classfile.AttributeOwnerComponent
```

#### FIELDS

---

- private short descriptorIndex
  - The descriptor index

CONSTRUCTORS

---

• *MethodInfo*

```
public MethodInfo( koala.dynamicjava.classfile.ConstantPool cp,
    java.lang.String rt, java.lang.String nm, java.lang.String [] pt )
```

– **Usage**

- \* Creates a new method info The type names must be fully qualified.

The following strings are valid class names:

```
. "int" </li>
. "Z" </li>
. "java.lang.String" </li>
. "java.lang.Object[]" </li>
. "[java/lang/String;" </li>
. "[[Ljava/lang/Integer;" </li>
```

– **Parameters**

- \* **cp** - the constant pool where constants are stored
- \* **rt** - the return type of this method
- \* **nm** - the name of this method
- \* **pt** - the parameters type names

METHODS

---

• *createCodeAttribute*

```
public CodeAttribute createCodeAttribute( )
```

– **Usage**

- \* Creates the code attribute for this method

• *createExceptionsAttribute*

```
public ExceptionsAttribute createExceptionsAttribute( )
```

– **Usage**

- \* Creates the exception attribute for this method

• *isAbstract*

```
public boolean isAbstract( )
```

– **Usage**

- \* Tests if the method is abstract

• *isStatic*

```
public boolean isStatic( )
```

– **Usage**

- \* Tests if the method is static

• *setAbstract*

```
public void setAbstract( )
```

– **Usage**

- \* Sets the abstract flag for this class



- 
- *setFinal*  
 public void **setFinal**( )  
 – **Usage**  
   \* Sets the final flag for this class

---

  - *setNative*  
 public void **setNative**( )  
 – **Usage**  
   \* Sets the native flag for this class

---

  - *setPrivate*  
 public void **setPrivate**( )  
 – **Usage**  
   \* Sets the private flag for this class

---

  - *setProtected*  
 public void **setProtected**( )  
 – **Usage**  
   \* Sets the protected flag for this class

---

  - *setPublic*  
 public void **setPublic**( )  
 – **Usage**  
   \* Sets the public flag for this class

---

  - *setSignature*  
 private void **setSignature**( java.lang.String rt, java.lang.String [] pt )  
 – **Usage**  
   \* Sets the signature of this method.  
 – **Parameters**  
   \* **rt** - the return type name  
   \* **pt** - the parameters type names

---

  - *setStatic*  
 public void **setStatic**( )  
 – **Usage**  
   \* Sets the static flag for this class

---

  - *setStrict*  
 public void **setStrict**( )  
 – **Usage**  
   \* Sets the strict flag for this class

---

- *setSynchronized*  
 public void **setSynchronized**( )  
 – **Usage**  
   \* Sets the synchronized flag for this class  


---
- *write*  
 public void **write**( java.io.DataOutputStream out )  
 – **Usage**  
   \* Writes the method info to the given output stream

---

 METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.AttributeOwnerComponent
 

---

( in 14.1.3, page 1045)

- *setAccessFlags*  
 public void **setAccessFlags**( int flags )  
 – **Usage**  
   \* Sets the access flags for this class  


---
- *setSyntheticAttribute*  
 public void **setSyntheticAttribute**( )  
 – **Usage**  
   \* Sets the synthetic attribute to this field

---

 METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.BytecodeComponent
 

---

( in 14.1.4, page 1046)

- *getConstantPool*  
 public ConstantPool **getConstantPool**( )  
 – **Usage**  
   \* Returns the constant pool  


---
- *write*  
 public abstract void **write**( java.io.DataOutputStream out )  
 – **Usage**  
   \* Writes the class file to the given output stream  


---
- *write*  
 public void **write**( java.io.OutputStream out )  
 – **Usage**  
   \* Writes the class file to the given output stream

### 14.1.35 CLASS SimpleAttribute

---

This class represents an attribute that is composed of his name

DECLARATION

---

```
public class SimpleAttribute
extends koala.dynamicjava.classfile.AttributeInfo
```

CONSTRUCTORS

---

- *SimpleAttribute*  

```
public SimpleAttribute( koala.dynamicjava.classfile.ConstantPool  cp,
java.lang.String  nm )
```

  - **Usage**
    - \* Creates a new attribute
  - **Parameters**
    - \* **cp** - the constant pool
    - \* **nm** - the attribute name

METHODS

---

- *write*  

```
public void write( java.io.DataOutputStream  out )
```

  - **Usage**
    - \* Writes the exception info to the given output stream.

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.AttributeInfo

---

( in 14.1.2, page 1044)

METHODS INHERITED FROM CLASS koala.dynamicjava.classfile.BytecodeComponent

---

( in 14.1.4, page 1046)

- *getConstantPool*  

```
public ConstantPool getConstantPool( )
```

  - **Usage**
    - \* Returns the constant pool
- *write*  

```
public abstract void write( java.io.DataOutputStream  out )
```

  - **Usage**
    - \* Writes the class file to the given output stream
- *write*  

```
public void write( java.io.OutputStream  out )
```

  - **Usage**
    - \* Writes the class file to the given output stream

## Chapter 15

### Package

### koala.dynamicjava.interpreter.modifier

*Package Contents*

*Page*

---

#### Classes

<b>ArrayModifier</b> .....	1092
<i>This interface represents objets that modify an array</i>	
<b>FinalVariableModifier</b> .....	1093
<i>This interface represents objets that modify a final variable</i>	
<b>InvalidModifier</b> .....	1094
<i>This interface represents an invalid modifier</i>	
<b>LeftHandSideModifier</b> .....	1095
<i>This class represents the objets that modify the left hand side of an assignment.</i>	
<b>ObjectFieldModifier</b> .....	1096
<i>This interface represents the objets that modify an object field</i>	
<b>StaticFieldModifier</b> .....	1097
<i>This interface represents the objets that modify a field</i>	
<b>SuperFieldModifier</b> .....	1098
<i>This interface represents the objets that modify an super field</i>	
<b>VariableModifier</b> .....	1099
<i>This interface represents objets that modify a variable</i>	

---

## 15.1 Classes

### 15.1.1 CLASS ArrayModifier

This interface represents objects that modify an array

#### DECLARATION

```
public class ArrayModifier
extends koala.dynamicjava.interpreter.modifier.LeftHandSideModifier
```

#### CONSTRUCTORS

- *ArrayModifier*  
 public **ArrayModifier**( koala.dynamicjava.tree.ArrayAccess node )
  - **Usage**
    - \* Creates a new array modifier
  - **Parameters**
    - \* **node** - the node of that represents this array

#### METHODS

- *modify*  
 public void **modify**( koala.dynamicjava.interpreter.context.Context ctx,  
 java.lang.Object value )
  - **Usage**
    - \* Sets the value of the underlying left hand side expression
- *prepare*  
 public Object **prepare**( koala.dynamicjava.tree.visitor.Visitor v,  
 koala.dynamicjava.interpreter.context.Context ctx )
  - **Usage**
    - \* Prepares the modifier for modification

#### METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.modifier.LeftHandSideModifier

( in 15.1.4, page 1095)

- *modify*  
 public abstract void **modify**( koala.dynamicjava.interpreter.context.Context ctx,  
 java.lang.Object value )
  - **Usage**
    - \* Sets the value of the underlying left hand side expression

- 
- *prepare*  

```
public abstract Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
                               koala.dynamicjava.interpreter.context.Context ctx )
```

    - **Usage**  
 \* Prepares the modifier for modification
    - **Returns** - the value of the left hand side

### 15.1.2 CLASS *FinalVariableModifier*

---

This interface represents objects that modify a final variable

#### DECLARATION

---

```
public class FinalVariableModifier
extends koala.dynamicjava.interpreter.modifier.VariableModifier
```

#### CONSTRUCTORS

---

- *FinalVariableModifier*  

```
public FinalVariableModifier( koala.dynamicjava.tree.QualifiedName name,
                              java.lang.Class type )
```

  - **Usage**  
 \* Creates a new final variable modifier
  - **Parameters**  
 \* **name** - the node of that represents this variable  
 \* **type** - the declared type of the variable

#### METHODS

---

- *modify*  

```
public void modify( koala.dynamicjava.interpreter.context.Context ctx,
                    java.lang.Object value )
```

  - **Usage**  
 \* Sets the value of the underlying left hand side expression

#### METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.modifier.VariableModifier

---

( in 15.1.8, page 1099)

- *modify*  

```
public void modify( koala.dynamicjava.interpreter.context.Context ctx,
                    java.lang.Object value )
```

  - **Usage**

- \* Sets the value of the underlying left hand side expression
- *prepare*  

```
public Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
koala.dynamicjava.interpreter.context.Context ctx )
```

  - **Usage**
    - \* Prepares the modifier for modification

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.modifier.LeftHandSideModifier

( in 15.1.4, page 1095)

- *modify*  

```
public abstract void modify( koala.dynamicjava.interpreter.context.Context ctx,
java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of the underlying left hand side expression
- *prepare*  

```
public abstract Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
koala.dynamicjava.interpreter.context.Context ctx )
```

  - **Usage**
    - \* Prepares the modifier for modification
  - **Returns** - the value of the left hand side

**15.1.3 CLASS InvalidModifier**

This interface represents an invalid modifier

## DECLARATION

```
public class InvalidModifier
extends koala.dynamicjava.interpreter.modifier.LeftHandSideModifier
```

## FIELDS

- private Node node
  - The node

## CONSTRUCTORS

- *InvalidModifier*  

```
public InvalidModifier( koala.dynamicjava.tree.Node n )
```

  - **Usage**
    - \* Creates a new field modifier
  - **Parameters**
    - \* **n** - the node

## METHODS

- *modify*  

```
public void modify( koala.dynamicjava.interpreter.context.Context ctx,
    java.lang.Object value )
```

    - **Usage**  
 \* Sets the value of the underlying left hand side expression
- 
- *prepare*  

```
public Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
    koala.dynamicjava.interpreter.context.Context ctx )
```

    - **Usage**  
 \* Prepares the modifier for modification

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.modifier.LeftHandSideModifier

( in 15.1.4, page 1095)

- *modify*  

```
public abstract void modify( koala.dynamicjava.interpreter.context.Context ctx,
    java.lang.Object value )
```

    - **Usage**  
 \* Sets the value of the underlying left hand side expression
- 
- *prepare*  

```
public abstract Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
    koala.dynamicjava.interpreter.context.Context ctx )
```

    - **Usage**  
 \* Prepares the modifier for modification
    - **Returns** - the value of the left hand side

## 15.1.4 CLASS LeftHandSideModifier

This class represents the objects that modify the left hand side of an assignment.

## DECLARATION

```
public abstract class LeftHandSideModifier
extends java.lang.Object
```

## CONSTRUCTORS

- *LeftHandSideModifier*  

```
public LeftHandSideModifier( )
```



METHODS

---

• *modify*

```
public abstract void modify( koala.dynamicjava.interpreter.context.Context
                             ctx, java.lang.Object  value )
```

– **Usage**

\* Sets the value of the underlying left hand side expression

---

• *prepare*

```
public abstract Object prepare( koala.dynamicjava.tree.visitor.Visitor  v,
                                 koala.dynamicjava.interpreter.context.Context  ctx )
```

– **Usage**

\* Prepares the modifier for modification

– **Returns** - the value of the left hand side**15.1.5 CLASS ObjectFieldModifier**

---

This interface represents the objects that modify an object field

DECLARATION

---

```
public class ObjectFieldModifier
extends koala.dynamicjava.interpreter.modifier.LeftHandSideModifier
```

CONSTRUCTORS

---

• *ObjectFieldModifier*

```
public ObjectFieldModifier( java.lang.reflect.Field  f,
                             koala.dynamicjava.tree.ObjectFieldAccess  n )
```

– **Usage**

\* Creates a new field modifier

– **Parameters**

\* **f** - the field to modify

\* **n** - the field access node

METHODS

---

• *modify*

```
public void modify( koala.dynamicjava.interpreter.context.Context  ctx,
                    java.lang.Object  value )
```

– **Usage**

\* Sets the value of the underlying left hand side expression

---

- *prepare*

```
public Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
    koala.dynamicjava.interpreter.context.Context ctx )
```

- **Usage**

- \* Prepares the modifier for modification

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.modifier.LeftHandSideModifier

( in 15.1.4, page 1095)

- *modify*

```
public abstract void modify( koala.dynamicjava.interpreter.context.Context ctx,
    java.lang.Object value )
```

- **Usage**

- \* Sets the value of the underlying left hand side expression

---

- *prepare*

```
public abstract Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
    koala.dynamicjava.interpreter.context.Context ctx )
```

- **Usage**

- \* Prepares the modifier for modification

- **Returns** - the value of the left hand side

### 15.1.6 CLASS StaticFieldModifier

---

This interface represents the objects that modify a field

## DECLARATION

```
public class StaticFieldModifier
extends koala.dynamicjava.interpreter.modifier.LeftHandSideModifier
```

## CONSTRUCTORS

- *StaticFieldModifier*

```
public StaticFieldModifier( java.lang.reflect.Field f,
    koala.dynamicjava.tree.Node n )
```

- **Usage**

- \* Creates a new field modifier

- **Parameters**

- \* **f** - the field to modify
  - \* **n** - the field access node

## METHODS

- 
- *modify*  
 public void **modify**( koala.dynamicjava.interpreter.context.Context ctx,  
 java.lang.Object value )  
 – **Usage**  
 \* Sets the value of the underlying left hand side expression

---

  - *prepare*  
 public Object **prepare**( koala.dynamicjava.tree.visitor.Visitor v,  
 koala.dynamicjava.interpreter.context.Context ctx )  
 – **Usage**  
 \* Prepares the modifier for modification

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.modifier.LeftHandSideModifier

( in 15.1.4, page 1095)

- 
- *modify*  
 public abstract void **modify**( koala.dynamicjava.interpreter.context.Context ctx,  
 java.lang.Object value )  
 – **Usage**  
 \* Sets the value of the underlying left hand side expression

---

  - *prepare*  
 public abstract Object **prepare**( koala.dynamicjava.tree.visitor.Visitor v,  
 koala.dynamicjava.interpreter.context.Context ctx )  
 – **Usage**  
 \* Prepares the modifier for modification  
 – **Returns** - the value of the left hand side

## 15.1.7 CLASS SuperFieldModifier

---

This interface represents the objects that modify an super field

## DECLARATION

---

public class SuperFieldModifier <b>extends</b> koala.dynamicjava.interpreter.modifier.LeftHandSideModifier
---

CONSTRUCTORS

---

• *SuperFieldModifier*

```
public SuperFieldModifier( java.lang.reflect.Field f,
koala.dynamicjava.tree.SuperFieldAccess n )
```

## – Usage

\* Creates a new field modifier

## – Parameters

\* **f** - the field to modify  
 \* **n** - the field access node

METHODS

---

• *modify*

```
public void modify( koala.dynamicjava.interpreter.context.Context ctx,
java.lang.Object value )
```

## – Usage

\* Sets the value of the underlying left hand side expression

• *prepare*

```
public Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
koala.dynamicjava.interpreter.context.Context ctx )
```

## – Usage

\* Prepares the modifier for modification

## METHODS INHERITED FROM CLASS

koala.dynamicjava.interpreter.modifier.LeftHandSideModifier

---

( in 15.1.4, page 1095)

• *modify*

```
public abstract void modify( koala.dynamicjava.interpreter.context.Context ctx,
java.lang.Object value )
```

## – Usage

\* Sets the value of the underlying left hand side expression

• *prepare*

```
public abstract Object prepare( koala.dynamicjava.tree.visitor.Visitor v,
koala.dynamicjava.interpreter.context.Context ctx )
```

## – Usage

\* Prepares the modifier for modification

## – Returns - the value of the left hand side

**15.1.8 CLASS VariableModifier**

---

This interface represents objects that modify a variable

DECLARATION

---

```
public class VariableModifier
extends koala.dynamicjava.interpreter.modifier.LeftHandSideModifier
```

CONSTRUCTORS

---

- *VariableModifier*  

```
public VariableModifier( koala.dynamicjava.tree.QualifiedName  name,
java.lang.Class  type )
```

  - **Usage**
    - \* Creates a new variable modifier
  - **Parameters**
    - \* **name** - the node of that represents this variable
    - \* **type** - the declared type of the variable

METHODS

---

- *modify*  

```
public void modify( koala.dynamicjava.interpreter.context.Context  ctx,
java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of the underlying left hand side expression
- 
- *prepare*  

```
public Object prepare( koala.dynamicjava.tree.visitor.Visitor  v,
koala.dynamicjava.interpreter.context.Context  ctx )
```

    - **Usage**
      - \* Prepares the modifier for modification

## METHODS INHERITED FROM CLASS

```
koala.dynamicjava.interpreter.modifier.LeftHandSideModifier
```

---

( in 15.1.4, page 1095)

- *modify*  

```
public abstract void modify( koala.dynamicjava.interpreter.context.Context  ctx,
java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of the underlying left hand side expression
- 
- *prepare*  

```
public abstract Object prepare( koala.dynamicjava.tree.visitor.Visitor  v,
koala.dynamicjava.interpreter.context.Context  ctx )
```

    - **Usage**
      - \* Prepares the modifier for modification
    - **Returns** - the value of the left hand side

## Chapter 16

# Package jeliot.theater

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Interfaces</b>	
<b>ActorContainer</b> .....	1104
<i>ActoContainer interface is implemented by classes that are going to contain Actors as their fields and take care of their painting.</i>	
<b>Controlled</b> .....	1104
<i>Controlled interface is implemented by a class whose instances should be controlled by the ThreadController instance.</i>	
<b>Classes</b>	
<b>ACActor</b> .....	1105
<i>Array Creation actor shows the "new Type[n]" is shown before the array is created.</i>	
<b>Actor</b> .....	1109
<i>The base class for all the actors.</i>	
<b>ActorFactory</b> .....	1115
<i>This class handles the centralized creation of the actors.</i>	
<b>AnimatingActor</b> .....	1122
<i>Animating actor is a actor that may be used to show frame based animation.</i>	
<b>Animation</b> .....	1127
<i>Animation class represents one atomic animation in Jeliot.</i>	
<b>AnimationEngine</b> .....	1130
<i>AnimationEngine schedules the animations represented by instances of Animation class.</i>	
<b>Animator</b> .....	1132
<i>An abstract class that handles the animation.</i>	
<b>ArrayActor</b> .....	1134
<i>Array Actor represents the array instance.</i>	
<b>BubbleActor</b> .....	1139
<i>Bubble actor is used to move the return value from the method stage to the scratch (evaluation area).</i>	
<b>CastActor</b> .....	1144
<i>CastActor handles the animation of the casting of the primitive values.</i>	
<b>ComponentDragger</b> .....	1149
<i>This class is not currently used by Jeliot.</i>	

<b>ConstantBox</b> .....	1150
<i>Constant box instance represents a place where all the literal constants appear during the animation.</i>	
<b>Director</b> .....	1154
<i>Directs the program animation.</i>	
<b>Director.InputAnimator</b> .....	1168
<i>...no description...</i>	
<b>ExpressionActor</b> .....	1169
<i>ExpressionActor represents a single line of a scratch the evaluation area.</i>	
<b>Highlight</b> .....	1175
<i>Highlight represents a single highlight of the source code during the animation.</i>	
<b>ImageLoader</b> .....	1176
<i>This class handles the image loading and caching for the animation.</i>	
<b>ImageValueActor</b> .....	1177
<i>ImageValueActor is an actor that is used when a value actor should be an image.</i>	
<b>IndexActor</b> .....	1182
<i>IndexActor shows the line between the array access' indexing expression result value and the array's actual index.</i>	
<b>InputComponent</b> .....	1187
<i>InputComponent is shown when ever the executed program requests input.</i>	
<b>InputValidator</b> .....	1211
<i>InputValidator is the base class for all the input validators for the input component.</i>	
<b>InstanceActor</b> .....	1212
<i>InstanceActor is a base class for all the instances: ArrayActors and ObjectStage.</i>	
<b>LinesAndText</b> .....	1217
<i>This actor draws the dashed lines and titles to separate explicitly the theater (animation frame) into four areas: constant area, method area, object and array area and expression evaluation area.</i>	
<b>MessageActor</b> .....	1223
<i>MessageActor shows all the textual messages to the user.</i>	
<b>MethodStage</b> .....	1228
<i>MethodStage is the graphical representation of the MethodFrame.</i>	
<b>ObjectStage</b> .....	1234
<i>MethodStage is the graphical representation of the ObjectFrame.</i>	
<b>OMIActor</b> .....	1241
<i>OMIActor represents graphically the object method invocation.</i>	
<b>OperatorActor</b> .....	1246
<i>An instance of the OperatorActor class represents a operator in the expressions.</i>	
<b>PanelActor</b> .....	1250
<i>PanelActor represents the curtains of the theater and produces the opening and closing animations of the curtains.</i>	
<b>PanelController</b> .....	1256
<i>PanelController handles the curtains of the theater (PanelActor) by controlling the opening and closing of the curtains and showing the panel and background images.</i>	
<b>ReferenceActor</b> .....	1257

<i>ReferenceActor shows the reference to some InstanceActor (e.g.</i>	
<b>ReferenceVariableActor</b> .....	1264
<i>ReferenceVariableActor represents graphically the variables of the reference type.</i>	
<b>Scratch</b> .....	1270
<i>Scratch controls the expression evaluation area.</i>	
<b>SMIActor</b> .....	1276
<i>SMIActor represents graphically the static method invocation.</i>	
<b>Theater</b> .....	1280
<i>This is the theatre component that is added in the left pane of the user interface and on which the program animation produced in the theater package is currently drawn.</i>	
<b>TheaterManager</b> .....	1307
<i>TheaterManager allocates the space for all InstanceActors, MethodStages, Scratches and constants (ConstantBox), and also listens the Theater component for resizes so the the allocation of the space is valid after resizing of the Theater component.</i>	
<b>ThreadController</b> .....	1311
<i>ThreadController allows the execution of the Runnable object controlled by it to be paused and resumed in a safe way.</i>	
<b>Trace</b> .....	1312
<i>The trace that an actor leaves when moving.</i>	
<b>ValueActor</b> .....	1313
<i>ValueActor is an actor that represents graphically the language construct Value.</i>	
<b>VariableActor</b> .....	1318
<i>VariableActor represent graphically the language construct Variable for primitive data types and Strings.</i>	
<b>VariableInArrayActor</b> .....	1323
<i>VariableInArrayActor represent graphically the language construct VariableInArray for primitive data types and Strings.</i>	

---



## 16.1 Interfaces

### 16.1.1 INTERFACE ActorContainer

---

**ActoContainer** interface is implemented by classes that are going to contain **Actors** as their fields and take care of their painting. This means that those classes having actors as their fields but not taking care of the painting of the actors are not implementing this class.

#### DECLARATION

---

```
public interface ActorContainer
```

#### METHODS

---

- *removeActor*  

```
public void removeActor( jeliot.theater.Actor actor )
```

  - **Usage**
    - \* Remove the the given actor from the ActorContainer.
  - **Parameters**
    - \* **actor** - The actor to be removed.

### 16.1.2 INTERFACE Controlled

---

**Controlled** interface is implemented by a class whose instances should be controlled by the **ThreadController** instance.

#### DECLARATION

---

```
public interface Controlled
```

#### METHODS

---

- *resume*  

```
public void resume( )
```

  - **Usage**
    - \* Continue the action that was suspended.

---
- *suspend*  

```
public void suspend( )
```

  - **Usage**
    - \* Suspend the action that is currently done.

## 16.2 Classes

### 16.2.1 CLASS ACActor

---

Array Creation actor shows the "new Type[n]" is shown before the array is created. The structure is similar to SMIActor.

#### DECLARATION

---

```
public class ACActor
extends jeliot.theater.Actor
implements ActorContainer
```

#### CONSTRUCTORS

---

- *ACActor*  
 public **ACActor**( java.lang.String name, int n )  
   – **Parameters**  
     \* name -  
     \* n -

#### METHODS

---

- *bind*  
 public void **bind**( jeliot.theater.Actor actor )  
   – **Parameters**  
     \* actor -

---

- *calculateSize*  
 public void **calculateSize**( )

---

- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )

---

- *paintActors*  
 public void **paintActors**( java.awt.Graphics g )  
   – **Parameters**  
     \* g -

---

- *removeActor*  
 public void **removeActor**( jeliot.theater.Actor actor )

---

- *reserve*  
 public Point **reserve**( jeliot.theater.Actor actor )  
   – **Parameters**

\* actor -  
 – Returns -

---

- *setLight*  
 public void **setLight**( int light )

#### METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
 public Animation **appear**( java.awt.Point loc )  
 – Usage  
 \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.  
 – Parameters  
 \* loc -  
 – Returns -
- *calculateSize*  
 public void **calculateSize**( )
- *clone*  
 public Object **clone**( )
- *fly*  
 public Animation **fly**( java.awt.Point p )  
 – Parameters  
 \* p -  
 – Returns -
- *fly*  
 public Animation **fly**( java.awt.Point p, int shadow )  
 – Usage  
 \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.  
 – Parameters  
 \* p -  
 \* shadow -  
 – Returns -
- *getActorAt*  
 public Actor **getActorAt**( int x, int y )  
 – Parameters  
 \* x -  
 \* y -  
 – Returns -
- *getBackground*  
 public Color **getBackground**( )  
 – Returns -
- *getFont*  
 public Font **getFont**( )

- 
- **Returns** -

---

  - *getFontMetrics*  
protected FontMetrics **getFontMetrics**( )
  - **Returns** -

---

  - *getForeground*  
public Color **getForeground**( )
  - **Returns** -

---

  - *getHeight*  
public int **getHeight**( )
  - **Returns** -

---

  - *getLocation*  
public Point **getLocation**( )
  - **Returns** -

---

  - *getParent*  
public ActorContainer **getParent**( )
  - **Returns** -

---

  - *getRootLocation*  
public Point **getRootLocation**( )
  - **Returns** -

---

  - *getShadow*  
public int **getShadow**( )
  - **Returns** -

---

  - *getSize*  
public Dimension **getSize**( )
  - **Returns** -

---

  - *getWidth*  
public int **getWidth**( )
  - **Returns** -

---

  - *getX*  
public int **getX**( )
  - **Returns** -

---

  - *getY*  
public int **getY**( )
  - **Returns** -

---

  - *paintActor*  
public void **paintActor**( java.awt.Graphics g )
  - **Usage**  
\* Paints the actor on the given Graphics instance. Override this in subclasses.
  - **Parameters**  
\* g - The Graphics object

---

- *paintActors*

```
protected void paintActors( java.awt.Graphics g, java.util.Vector actors )
```

- Usage

- \* Paints the actors contained in the vector on this actor.

- Parameters

- \* g -
    - \* actors -

---

- *paintBackground*

```
protected void paintBackground( java.awt.Graphics g, java.awt.Image backImage,
int xx, int yy, int w, int h )
```

- Parameters

- \* g -
    - \* backImage -
    - \* xx -
    - \* yy -
    - \* w -
    - \* h -

---

- *paintShadow*

```
public void paintShadow( java.awt.Graphics g )
```

- Usage

- \* Paints the shadow of the actor. Override this in the subclasses if needed.

- Parameters

- \* g -

---

- *setBackground*

```
public void setBackground( java.awt.Color bgcolor )
```

- Parameters

- \* bgcolor -

---

- *setBorderWidth*

```
public void setBorderWidth( int w )
```

- Parameters

- \* w -

---

- *setBounds*

```
public void setBounds( int x, int y, int w, int h )
```

- Parameters

- \* x -
    - \* y -
    - \* w -
    - \* h -

---

- *setFont*

```
public void setFont( java.awt.Font font )
```

- Parameters

- \* font -

---

- *setForeground*

```
public void setForeground( java.awt.Color fgcolor )
```

- Parameters

- \* fgcolor -

---

- *setInsets*  
 public void setInsets( java.awt.Insets insets )  
 – Parameters  
 \* insets -  


---
- *setLight*  
 public void setLight( int light )  
 – Parameters  
 \* light -  


---
- *setLocation*  
 public void setLocation( int x, int y )  
 – Parameters  
 \* x -  
 \* y -  


---
- *setLocation*  
 public void setLocation( java.awt.Point loc )  
 – Parameters  
 \* loc -  


---
- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )  
 – Parameters  
 \* parent -  


---
- *setShadow*  
 public void setShadow( int s )  
 – Parameters  
 \* s -  


---
- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )  
 – Parameters  
 \* si -  


---
- *setSize*  
 public void setSize( java.awt.Dimension d )  
 – Parameters  
 \* d -  


---
- *setSize*  
 public void setSize( int w, int h )  
 – Parameters  
 \* w -  
 \* h -

### 16.2.2 CLASS Actor

---

The base class for all the actors. The Actor class should not be used directly but indirectly with it's subclasses.

DECLARATION

---

```
public abstract class Actor
extends java.lang.Object
implements java.lang.Cloneable
```

FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for theater package.
- public static Component dummy
  -
- public static Image shadowImage
  -
- private static Font defaultFont
  -
- public static final int HIGHLIGHT
  -
- public static final int NORMAL
  -
- public static final int SHADED
  -
- private int x
  - Actor's x-coordinate in parent.
- private int y
  - Actor's y-coordinate in parent.
- private Trace trace
  - Trace left by the actor.
- private ActorContainer parent
  - Parent actor.

CONSTRUCTORS

---

- *Actor*  
**public Actor( )**

METHODS

---

• *appear*

```
public Animation appear( java.awt.Point loc )
```

## – Usage

\* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.

## – Parameters

\* loc -

– Returns -

---

• *calculateSize*

```
public void calculateSize( )
```

---

• *clone*

```
public Object clone( )
```

---

• *fly*

```
public Animation fly( java.awt.Point p )
```

## – Parameters

\* p -

– Returns -

---

• *fly*

```
public Animation fly( java.awt.Point p, int shadow )
```

## – Usage

\* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.

## – Parameters

\* p -

\* shadow -

– Returns -

---

• *getActorAt*

```
public Actor getActorAt( int x, int y )
```

## – Parameters

\* x -

\* y -

– Returns -

---

• *getBackground*

```
public Color getBackground( )
```

– Returns -

---

• *getFont*

```
public Font getFont( )
```



– Returns -

- 
- *getFontMetrics*  
protected FontMetrics getFontMetrics( )

– Returns -

- 
- *getForeground*  
public Color getForeground( )

– Returns -

- 
- *getHeight*  
public int getHeight( )

– Returns -

- 
- *getLocation*  
public Point getLocation( )

– Returns -

- 
- *getParent*  
public ActorContainer getParent( )

– Returns -

- 
- *getRootLocation*  
public Point getRootLocation( )

– Returns -

- 
- *getShadow*  
public int getShadow( )

– Returns -

- 
- *getSize*  
public Dimension getSize( )

– Returns -

- 
- *getWidth*  
public int getWidth( )

– Returns -

- 
- *getX*  
public int getX( )

– Returns -

- 
- *getY*  
public int getY( )

– Returns -

---

- *paintActor*

```
public void paintActor( java.awt.Graphics g )
```

- **Usage**

- \* Paints the actor on the given Graphics instance. Override this in subclasses.

- **Parameters**

- \* *g* - The Graphics object

---

- *paintActors*

```
protected void paintActors( java.awt.Graphics g, java.util.Vector actors )
```

- **Usage**

- \* Paints the actors contained in the vector on this actor.

- **Parameters**

- \* *g* -
    - \* *actors* -

---

- *paintBackground*

```
protected void paintBackground( java.awt.Graphics g, java.awt.Image  
backImage, int xx, int yy, int w, int h )
```

- **Parameters**

- \* *g* -
    - \* *backImage* -
    - \* *xx* -
    - \* *yy* -
    - \* *w* -
    - \* *h* -

---

- *paintShadow*

```
public void paintShadow( java.awt.Graphics g )
```

- **Usage**

- \* Paints the shadow of the actor. Override this in the subclasses if needed.

- **Parameters**

- \* *g* -

---

- *setBackground*

```
public void setBackground( java.awt.Color bgcolor )
```

- **Parameters**

- \* *bgcolor* -

---

- *setBorderWidth*

```
public void setBorderWidth( int w )
```

- **Parameters**

- \* *w* -

---

- *setBounds*

```
public void setBounds( int x, int y, int w, int h )
```

- **Parameters**

- \* x -
    - \* y -
    - \* w -
    - \* h -

---

- *setFont*

```
public void setFont( java.awt.Font font )
```

- **Parameters**

- \* font -

---

- *setForeground*

```
public void setForeground( java.awt.Color fgcolor )
```

- **Parameters**

- \* fgcolor -

---

- *setInsets*

```
public void setInsets( java.awt.Insets insets )
```

- **Parameters**

- \* insets -

---

- *setLight*

```
public void setLight( int light )
```

- **Parameters**

- \* light -

---

- *setLocation*

```
public void setLocation( int x, int y )
```

- **Parameters**

- \* x -
    - \* y -

---

- *setLocation*

```
public void setLocation( java.awt.Point loc )
```

- **Parameters**

- \* loc -

---

- *setParent*

```
public void setParent( jeliot.theater.ActorContainer parent )
```

- **Parameters**

- \* parent -

---

- *setShadow*

```
public void setShadow( int s )
```

- **Parameters**
  - \* **s** -

---
- *setShadowImage*
  - public static void **setShadowImage**( java.awt.Image **si** )
  - **Parameters**
  - \* **si** -

---
- *setSize*
  - public void **setSize**( java.awt.Dimension **d** )
  - **Parameters**
  - \* **d** -

---
- *setSize*
  - public void **setSize**( int **w**, int **h** )
  - **Parameters**
  - \* **w** -
  - \* **h** -

### 16.2.3 CLASS ActorFactory

---

This class handles the centralized creation of the actors. This enables the centralized appearance handling.

#### DECLARATION

---

```
public class ActorFactory
extends java.lang.Object
```

#### FIELDS

---

- private static ResourceBundle **bundle**
  - The resource bundle for theater package
- private Component **dummy**
  -
- private ImageLoader **iLoad**
  -
- private Image **shadowImage**
  -
- private Image **messageImage**
  -

- private Font valueFont  
—
- private Font variableFont  
—
- private Font stageFont  
—
- private Font messageFont  
—
- private Font indexFont  
—
- private Font SMIFont  
—
- private Font OMIFont  
—
- private Font ACFont  
—
- private Font LATFont  
—
- private int valueHeight  
—
- private Insets variableInsets  
—
- private int margin  
—
- private Color messagebc  
—
- private Color messagefc  
—
- private Color trueColor  
—
- private Color falseColor  
—
- private Color opColor  
—

- private Color methodStageColor  
—
- private Color objectStageColor  
—
- private Color SMIColor  
—
- private Color variableForegroundColor  
—
- private Color valueForegroundColor  
—
- private Color OMIColor  
—
- private Color ACColor  
—
- private Color bubbleColor  
—
- private Color LATForegroundColor  
—
- private Color LATBackgroundColor  
—
- private Color valColor  
—
- private Color varColor  
—
- private String binOpImageName  
—
- private String unaOpImageName  
—
- private static int typeValWidth  
—
- private static int typeWidth  
—
- private MessageFormat objectStageTitle  
—

CONSTRUCTORS

---

• *ActorFactory*

```
public ActorFactory( jeliot.theater.ImageLoader iLoad )
```

– **Parameters**

\* iLoad -

METHODS

---

• *getMaxMethodStageWidth*

```
public static int getMaxMethodStageWidth( )
```

– **Returns** -

---

• *getMaxObjectStageWidth*

```
public static int getMaxObjectStageWidth( )
```

– **Returns** -

---

• *getMaxTypeWidth*

```
public static int getMaxTypeWidth( )
```

– **Returns** -

---

• *getTypeValueWidth*

```
public static int getTypeValueWidth( int n )
```

– **Parameters**

\* n -

– **Returns** -

---

• *getTypeWidth*

```
public static int getTypeWidth( int n )
```

– **Parameters**

\* n -

– **Returns** -

---

• *produceACActor*

```
public ACActor produceACActor( java.lang.String name, int paramCount )
```

– **Usage**

\* Array Creator Actor

– **Parameters**

\* name -

\* paramCount -

– **Returns** -

---

- *produceArrayActor*  
 public ArrayActor produceArrayActor( jeliot.lang.ArrayInstance array )  
 – **Parameters**  
   \* array -  
 – **Returns** -  


---
- *produceBinOpActor*  
 public OperatorActor produceBinOpActor( int op )  
 – **Parameters**  
   \* op -  
 – **Returns** -  


---
- *produceBinOpResActor*  
 public OperatorActor produceBinOpResActor( int op )  
 – **Parameters**  
   \* op -  
 – **Returns** -  


---
- *produceBubble*  
 public BubbleActor produceBubble( jeliot.theater.Actor actor )  
 – **Parameters**  
   \* actor -  
 – **Returns** -  


---
- *produceConstantBox*  
 public ConstantBox produceConstantBox( )  
 – **Returns** -  


---
- *produceEllipsis*  
 public OperatorActor produceEllipsis( )  
 – **Returns** -  


---
- *produceHand*  
 public AnimatingActor produceHand( )  
 – **Returns** -  


---
- *produceImage*  
 public Image produceImage( java.lang.String iname )  
 – **Parameters**  
   \* iname -  
 – **Returns** -  


---
- *produceLinesAndText*  
 public LinesAndText produceLinesAndText( )



– **Returns** -

---

- *produceMessageActor*

**MessageActor produceMessageActor( java.lang.String [] text )**

– **Parameters**

\* **text** -

– **Returns** -

---

- *produceMethodStage*

**public MethodStage produceMethodStage( jeliot.lang.MethodFrame m )**

– **Parameters**

\* **m** -

– **Returns** -

---

- *produceObjectStage*

**public ObjectStage produceObjectStage( jeliot.lang.ObjectFrame m )**

– **Parameters**

\* **m** -

– **Returns** -

---

- *produceObjectVariableActor*

**public VariableActor produceObjectVariableActor( jeliot.lang.Variable v )**

– **Parameters**

\* **v** -

– **Returns** -

---

- *produceOMIActor*

**public OMIActor produceOMIActor( java.lang.String name, int paramCount )**

– **Usage**

\* Object Method Invocation Actor

– **Parameters**

\* **name** -

\* **paramCount** -

– **Returns** -

---

- *produceOperatorActor*

**public OperatorActor produceOperatorActor( java.awt.Image image )**

– **Parameters**

\* **image** -

– **Returns** -

---

- *produceReferenceActor*

**public ReferenceActor produceReferenceActor( jeliot.lang.Reference rf )**

– **Parameters**

- \* rf -
  - **Returns** -

---
- *produceReferenceActor*  
 public ReferenceActor produceReferenceActor( jeliot.theater.ReferenceActor cloneActor )
  - **Parameters**
    - \* cloneActor -
  - **Returns** -

---
- *produceSMIActor*  
 public SMIActor produceSMIActor( java.lang.String name, int paramCount )
  - **Usage**
    - \* Static Method Invocation Actor.
  - **Parameters**
    - \* name -
    - \* paramCount -
  - **Returns** -

---
- *produceUnaOpActor*  
 public OperatorActor produceUnaOpActor( int op )
  - **Parameters**
    - \* op -
  - **Returns** -

---
- *produceUnaOpResActor*  
 public OperatorActor produceUnaOpResActor( int op )
  - **Parameters**
    - \* op -
  - **Returns** -

---
- *produceValueActor*  
 public ValueActor produceValueActor( jeliot.lang.Value val )
  - **Parameters**
    - \* val -
  - **Returns** -

---
- *produceValueActor*  
 public ValueActor produceValueActor( jeliot.theater.ValueActor cloneActor )
  - **Parameters**
    - \* cloneActor -
  - **Returns** -

---

- *produceVariableActor*  
`public VariableActor produceVariableActor( jeliot.lang.Variable v )`  
  - **Parameters**  
    - \* v -
  - **Returns** -

---
- *setStageFont*  
`public void setStageFont( java.awt.Font font )`  
  - **Parameters**  
    - \* font -

---
- *setValueFont*  
`public void setValueFont( java.awt.Font font )`  
  - **Parameters**  
    - \* font -

---
- *setVariableFont*  
`public void setVariableFont( java.awt.Font font )`  
  - **Parameters**  
    - \* font -

---

## 16.2.4 CLASS AnimatingActor

---

Animating actor is a actor that may be used to show frame based animation.

### DECLARATION

---

```
public class AnimatingActor
extends jeliot.theater.Actor
```

### FIELDS

---

- private Image image  
  -

### CONSTRUCTORS

---

- *AnimatingActor*  
`public AnimatingActor( java.awt.Image image )`  
  - **Parameters**  
    - \* image -

METHODS

---

- *calculateSize*  
public void calculateSize( )
  - *changeImage*  
 public Animation **changeImage**( java.awt.Image **chim** )
    - **Usage**
      - \* Returns an animation object that changes the image of this actor to the given image.
    - **Parameters**
      - \* **chim** -
    - **Returns** -
- 
- *paintActor*  
public void paintActor( java.awt.Graphics **g** )
  - *setImage*  
 public void **setImage**( java.awt.Image **image** )
    - **Parameters**
      - \* **image** -

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
 public Animation **appear**( java.awt.Point **loc** )
    - **Usage**
      - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
    - **Parameters**
      - \* **loc** -
    - **Returns** -
- 
- *calculateSize*  
public void calculateSize( )
  - *clone*  
public Object clone( )
  - *fly*  
 public Animation **fly**( java.awt.Point **p** )
    - **Parameters**
      - \* **p** -
    - **Returns** -
- 
- *fly*  
 public Animation **fly**( java.awt.Point **p**, int **shadow** )
    - **Usage**

\* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.

– **Parameters**

\* **p** -  
\* **shadow** -

– **Returns** -

---

- *getActorAt*

public Actor **getActorAt**( int **x**, int **y** )

– **Parameters**

\* **x** -  
\* **y** -

– **Returns** -

---

- *getBackground*

public Color **getBackground**( )

– **Returns** -

---

- *getFont*

public Font **getFont**( )

– **Returns** -

---

- *getFontMetrics*

protected FontMetrics **getFontMetrics**( )

– **Returns** -

---

- *getForeground*

public Color **getForeground**( )

– **Returns** -

---

- *getHeight*

public int **getHeight**( )

– **Returns** -

---

- *getLocation*

public Point **getLocation**( )

– **Returns** -

---

- *getParent*

public ActorContainer **getParent**( )

– **Returns** -

---

- *getRootLocation*

public Point **getRootLocation**( )

– **Returns** -

---

- *getShadow*

public int **getShadow**( )

– **Returns** -

---

- *getSize*

public Dimension **getSize**( )

– **Returns** -

---

- *getWidth*  
`public int getWidth( )`  
 – Returns -  


---
- *getX*  
`public int getX( )`  
 – Returns -  


---
- *getY*  
`public int getY( )`  
 – Returns -  


---
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`  
 – Usage  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – Parameters  
   \* g - The Graphics object  


---
- *paintActors*  
`protected void paintActors( java.awt.Graphics g, java.util.Vector actors )`  
 – Usage  
   \* Paints the actors contained in the vector on this actor.  
 – Parameters  
   \* g -  
   \* actors -  


---
- *paintBackground*  
`protected void paintBackground( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )`  
 – Parameters  
   \* g -  
   \* backImage -  
   \* xx -  
   \* yy -  
   \* w -  
   \* h -  


---
- *paintShadow*  
`public void paintShadow( java.awt.Graphics g )`  
 – Usage  
   \* Paints the shadow of the actor. Override this in the subclasses if needed.  
 – Parameters  
   \* g -  


---
- *setBackground*  
`public void setBackground( java.awt.Color bgcolor )`  
 – Parameters  
   \* bgcolor -  


---
- *setBorderWidth*  
`public void setBorderWidth( int w )`  
 – Parameters

- 
- \* w -

---

    - *setBounds*  
public void **setBounds**( int x, int y, int w, int h )  
      - Parameters
      - \* x -
      - \* y -
      - \* w -
      - \* h -

---
  - *setFont*  
public void **setFont**( java.awt.Font font )  
      - Parameters
      - \* font -

---
  - *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )  
      - Parameters
      - \* fgcolor -

---
  - *setInsets*  
public void **setInsets**( java.awt.Insets insets )  
      - Parameters
      - \* insets -

---
  - *setLight*  
public void **setLight**( int light )  
      - Parameters
      - \* light -

---
  - *setLocation*  
public void **setLocation**( int x, int y )  
      - Parameters
      - \* x -
      - \* y -

---
  - *setLocation*  
public void **setLocation**( java.awt.Point loc )  
      - Parameters
      - \* loc -

---
  - *setParent*  
public void **setParent**( jeliot.theater.ActorContainer parent )  
      - Parameters
      - \* parent -

---
  - *setShadow*  
public void **setShadow**( int s )  
      - Parameters
      - \* s -

---
  - *setShadowImage*  
public static void **setShadowImage**( java.awt.Image si )  
      - Parameters

---

- \* `si` -

---

  - *setSize*  
`public void setSize( java.awt.Dimension d )`
    - **Parameters**
    - \* `d` -

---

- *setSize*  
`public void setSize( int w, int h )`
  - **Parameters**
  - \* `w` -
  - \* `h` -

### 16.2.5 CLASS Animation

---

**Animation** class represents one atomic animation in **Jeliot**. Animation means here any event that includes movement of actors or that is otherwise dependent of time. Examples of animation include moving an actor from one place to another or flashing the colors of an actor as it is introduced to the theatre.

The animation is played by an instance of **AnimationEngine** class. The animation engine takes care of scheduling the animation. Animation class is the abstract superclass of various specialized animation classes. These subclasses must implement the **animate** method in which they make their changes to actors. The animation engine calls this method at even time intervals. If the animation has to do something in prior to starting the animation, especially if it has to set any parameters that depend on the duration of the animation or it has to add any actors to the theatre, it may do this in its **init()** method. When the animation finishes, the engine calls its **finish** method.

#### DECLARATION

---

```
public abstract class Animation
extends java.lang.Object
```

#### FIELDS

---

- private boolean `finished`
  - This flag is set by the `doFinish()` method when the animation is at end.
- private Theater `theatre`
  - The theatre in which the animation takes place.
- private int `startTime`
  - The starting time for the animation, in milliseconds.
- private int `duration`
  - Desired duration of the animation, in milliseconds.



- private Actor actor
  - An actor associated with this animation – the actor that gets animated.

## CONSTRUCTORS

---

- *Animation*  
public **Animation**( )

## METHODS

---

- *addActor*  
protected void **addActor**( jeliot.theater.Actor actor )
  - **Usage**
    - \* Adds an actor to the theatre in which the animation is performed. Called by subclasses.
  - **Parameters**
    - \* actor -

---
- *animate*  
public abstract void **animate**( double p )
  - **Usage**
    - \* This method performs the animation. It is abstract, so it must be implemented by the subclasses.
  - **Parameters**
    - \* p - Amount of work to do in this animation step. P of 1000 Measured in milliseconds; the animation will get roughly 1000 units of work per second if the animation engine is in its default configuration.

---
- *doFinish*  
public void **doFinish**( )
  - **Usage**
    - \* Finishes the animation.

---
- *finalFinish*  
public void **finalFinish**( )
  - **Usage**
    - \* This method is called when the animation is finished. Should be overridden.

---
- *finish*  
protected void **finish**( )
  - **Usage**
    - \* Finishes the animation. This method may be overridden by subclasses to do something after the animation has finished.

---

- *getDuration*

**public int getDuration( )**

- **Usage**

\* Returns the desired duration of this animation in milliseconds.

- **Returns** - the desired duration of this animation in milliseconds.

---

- *getStartTime*

**public int getStartTime( )**

- **Usage**

\* Returns the starting time of this animation in milliseconds.

- **Returns** - the starting time of this animation in milliseconds.

---

- *init*

**public void init( )**

- **Usage**

\* Initializes the animation. This method is called by the animation engine before starting the animation, after duration and theatre have been set. To be overridden by subclasses; default implementation does nothing.

---

- *isFinished*

**public boolean isFinished( )**

- **Usage**

\* Returns true if the animation has already finished. That is, it has set its finished-flag.

- **Returns** - true if the animation has already finished. That is, it has set its finished-flag.

---

- *passivate*

**protected void passivate( jeliot.theater.Actor actor )**

- **Parameters**

\* actor -

---

- *removeActor*

**protected void removeActor( jeliot.theater.Actor actor )**

- **Parameters**

\* actor -

---

- *repaint*

**protected void repaint( )**

- **Usage**

\* Repaints the theatre. This method is called by subclasses between animation steps.

---

- *setDuration*

**public void setDuration( int duration )**

- **Usage**

- \* Sets the desired duration for this animation. The actual duration depends on the speed of the animation engine playing the animation.

- **Parameters**

- \* **duration** - The desired duration in milliseconds.
- 

- *setStartTime*

```
public void setStartTime( int  startTime )
```

- **Usage**

- \* Sets the starting time for this animation.

- **Parameters**

- \* **startTime** - The desired starting time in milliseconds.
- 

- *setTheatre*

```
public void setTheatre( jeliot.theater.Theater  theatre )
```

- **Usage**

- \* Sets the theatre in which this animation is played. This method is called by the animation engine before performing the animation.

- **Parameters**

- \* **theatre** -

## 16.2.6 CLASS AnimationEngine

---

**AnimationEngine** schedules the animations represented by instances of **Animation** class. The engine is given an animation or an array of animations, and it plays those animations. The speed and quality of the animation can be controlled by setting its volume (speed) and FPS (Frames Per Second) values. An engine's volume is the amount of action it gives to the animation objects each second. The higher the volume, the faster the animations will play.

An animation engine may be assigned a **ThreadController** instance. In this case, the engine checks with the controller after every step of animation calling its **checkPoint** method.

### DECLARATION

---

```
public class AnimationEngine
extends java.lang.Object
implements Controlled
```

### FIELDS

---

- private double volume
  - Amount of action for one second period

- private double defaultVolume
  - Default amount of action for one second period
- private double fps
  - Number of times to act per second
- private double defaultFPS
  - Default number of times to act per second
- private boolean running
  - True if the animation engine is running.
- private Theater theatre
  - The theatre in which the animation takes place.
- private ThreadController controller
  - The thread controller of this animation engine. If controller null, it will be called after each step of animation.

## CONSTRUCTORS

---

- *AnimationEngine*  
 public **AnimationEngine**( jeliot.theater.Theater theatre )  
 – **Usage**  
 \* Constructs a new animation engine that will show its animations in given theatre.

## METHODS

---

- *resume*  
 public void **resume**( )  
 – **Usage**  
 \* Called by the thread controller when the animation is resumed.
- *setController*  
 public void **setController**( jeliot.theater.ThreadController controller )  
 – **Usage**  
 \* Sets the thread controller of this animation engine.
- *setDefaultValues*  
 public void **setDefaultValues**( )
- *setFPS*  
 public void **setFPS**( double fps )  
 – **Usage**  
 \* Sets the number of frames that should be shown in one second.

- *setVolume*  
`public void setVolume( double volume )`  
  - **Usage**  
 \* Sets the amount of action that is given to each animation at each step.

---
- *showAnimation*  
`public void showAnimation( jeliot.theater.Animation animation )`  
  - **Usage**  
 \* Performs the given animation.

---
- *showAnimation*  
`public void showAnimation( jeliot.theater.Animation [] animations )`  
  - **Usage**  
 \* Performs the animations in given array.

---
- *suspend*  
`public void suspend( )`  
  - **Usage**  
 \* Called by the thread controller when the animation is paused.

### 16.2.7 CLASS Animator

---

An abstract class that handles the animation. Currently, it is only used in the input handling in Theater class.

#### DECLARATION

---

```
public abstract class Animator
extends java.lang.Object
```

#### FIELDS

---

- private Value args  
 —
- private ValueActor argact  
 —
- private Value returnValue  
 —

CONSTRUCTORS

---

- *Animator*  
public **Animator**( )

METHODS

---

- *animate*  
public abstract void **animate**( jeliot.theater.Director director )  
 – **Parameters**  
   \* director -  


---
- *getArgument*  
public Value **getArgument**( int i )  
 – **Parameters**  
   \* i -  
 – **Returns** -  


---
- *getArgumentActor*  
protected Actor **getArgumentActor**( int i )  
 – **Parameters**  
   \* i -  
 – **Returns** -  


---
- *getReturnValue*  
public Value **getReturnValue**( )  
 – **Returns** -  


---
- *setArgumentActors*  
public void **setArgumentActors**( jeliot.theater.ValueActor [] argact )  
 – **Parameters**  
   \* argact -  


---
- *setArguments*  
public void **setArguments**( jeliot.lang.Value [] args )  
 – **Parameters**  
   \* args -  


---
- *setReturnValue*  
public void **setReturnValue**( jeliot.lang.Value v )  
 – **Parameters**  
   \* v -

### 16.2.8 CLASS *ArrayActor*

---

*Array Actor* represents the array instance.

#### DECLARATION

---

```
public class ArrayActor
extends jeliot.theater.InstanceActor
```

#### FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for theater package
- private String emptyArray1
  -
- private String emptyArray2
  -
- private Object variableActors
  -
- private Color valueColor
  -
- private int vlinex
  - The x-coordinate of the vertical line separating indices from values.
- private int valuw
  - The width of a cell reserved for a single value actor.
- private int valueh
  - The height of a single value actor.
- private int indexw
  - The width of an index label.
- private int dimensions
  -

#### CONSTRUCTORS

---

- *ArrayActor*

```
public ArrayActor( java.lang.Object  valueActors, int [] dimensions )
```

  - **Parameters**
    - \* **valueActors** -
    - \* **dimensions** -

METHODS

---

• *calculateSize*

```
public void calculateSize( int  valuw, int  valueh )
```

– **Parameters**

\* valuw -  
\* valueh -

---

• *getVariableActor*

```
public VariableActor getVariableActor( int [] index )
```

– **Parameters**

\* index -

– **Returns** -  

---

• *paintActor*

```
public void paintActor( java.awt.Graphics  g )
```

---

• *setValueColor*

```
public void setValueColor( java.awt.Color  valueColor )
```

– **Parameters**

\* valueColor -

METHODS INHERITED FROM CLASS jeliot.theater.InstanceActor

---

( in 16.2.22, page 1212)

• *addReference*

```
public void addReference( jeliot.theater.ReferenceActor  ref )
```

– **Parameters**

\* ref -

---

• *getNumberOfReferences*

```
public int getNumberOfReferences( )
```

– **Returns** -  

---

• *removeActor*

```
public void removeActor( jeliot.theater.Actor  actor )
```

---

• *removeReference*

```
public void removeReference( jeliot.theater.ReferenceActor  ref )
```

– **Parameters**

\* ref -

---

• *setLocation*

```
public void setLocation( int  x, int  y )
```



METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`  
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* `loc` -
  - **Returns** -

---
- *calculateSize*  
`public void calculateSize( )`  


---
- *clone*  
`public Object clone( )`  


---
- *fly*  
`public Animation fly( java.awt.Point p )`  
  - **Parameters**
    - \* `p` -
  - **Returns** -

---
- *fly*  
`public Animation fly( java.awt.Point p, int shadow )`  
  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* `p` -
    - \* `shadow` -
  - **Returns** -

---
- *getActorAt*  
`public Actor getActorAt( int x, int y )`  
  - **Parameters**
    - \* `x` -
    - \* `y` -
  - **Returns** -

---
- *getBackground*  
`public Color getBackground( )`  
  - **Returns** -

---
- *getFont*  
`public Font getFont( )`  
  - **Returns** -

---
- *getFontMetrics*  
`protected FontMetrics getFontMetrics( )`  
  - **Returns** -

---

- *getForeground*  
 public Color **getForeground**( )  
 – Returns -  


---
- *getHeight*  
 public int **getHeight**( )  
 – Returns -  


---
- *getLocation*  
 public Point **getLocation**( )  
 – Returns -  


---
- *getParent*  
 public ActorContainer **getParent**( )  
 – Returns -  


---
- *getRootLocation*  
 public Point **getRootLocation**( )  
 – Returns -  


---
- *getShadow*  
 public int **getShadow**( )  
 – Returns -  


---
- *getSize*  
 public Dimension **getSize**( )  
 – Returns -  


---
- *getWidth*  
 public int **getWidth**( )  
 – Returns -  


---
- *getX*  
 public int **getX**( )  
 – Returns -  


---
- *getY*  
 public int **getY**( )  
 – Returns -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – Usage  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – Parameters  
   \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – Usage  
   \* Paints the actors contained in the vector on this actor.  
 – Parameters  
   \* g -

- - \* actors -

---
- *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
    - Parameters
      - \* g -
      - \* backImage -
      - \* xx -
      - \* yy -
      - \* w -
      - \* h -

---
- *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )
    - Usage
      - \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - Parameters
      - \* g -

---
- *setBackground*  
public void **setBackground**( java.awt.Color bgcolor )
    - Parameters
      - \* bgcolor -

---
- *setBorderWidth*  
public void **setBorderWidth**( int w )
    - Parameters
      - \* w -

---
- *setBounds*  
public void **setBounds**( int x, int y, int w, int h )
    - Parameters
      - \* x -
      - \* y -
      - \* w -
      - \* h -

---
- *setFont*  
public void **setFont**( java.awt.Font font )
    - Parameters
      - \* font -

---
- *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )
    - Parameters
      - \* fgcolor -

---
- *setInsets*  
public void **setInsets**( java.awt.Insets insets )
    - Parameters
      - \* insets -

---
- *setLight*  
public void **setLight**( int light )

- **Parameters**
  - \* *light* -

---

- *setLocation*  
 public void setLocation( int x, int y )
  - **Parameters**
  - \* *x* -
  - \* *y* -

---

- *setLocation*  
 public void setLocation( java.awt.Point loc )
  - **Parameters**
  - \* *loc* -

---

- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* *parent* -

---

- *setShadow*  
 public void setShadow( int s )
  - **Parameters**
  - \* *s* -

---

- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - **Parameters**
  - \* *si* -

---

- *setSize*  
 public void setSize( java.awt.Dimension d )
  - **Parameters**
  - \* *d* -

---

- *setSize*  
 public void setSize( int w, int h )
  - **Parameters**
  - \* *w* -
  - \* *h* -

### 16.2.9 CLASS BubbleActor

---

Bubble actor is used to move the return value from the method stage to the scratch (evaluation area).

#### DECLARATION

---

```
public class BubbleActor
extends jeliot.theater.Actor
implements ActorContainer
```

CONSTRUCTORS

---

- *BubbleActor*  
 public **BubbleActor**( jeliot.theater.Actor speaker )  
 – **Parameters**  
 \* speaker -

METHODS

---

- *bind*  
 public void **bind**( )  


---
- *calculateSize*  
 public void **calculateSize**( )  


---
- *getActor*  
 public Actor **getActor**( )  
 – **Returns** -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  


---
- *paintActors*  
 public void **paintActors**( java.awt.Graphics g )  
 – **Parameters**  
 \* g -  


---
- *removeActor*  
 public void **removeActor**( jeliot.theater.Actor actor )  


---
- *removeTip*  
 public void **removeTip**( )  


---
- *reserve*  
 public Point **reserve**( jeliot.theater.Actor actor )  
 – **Parameters**  
 \* actor -  
 – **Returns** -  


---
- *setActor*  
 public void **setActor**( jeliot.theater.Actor actor )  
 – **Parameters**  
 \* actor -

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`  
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* `loc` -
  - **Returns** -

---

- *calculateSize*  
`public void calculateSize( )`

---

- *clone*  
`public Object clone( )`

---

- *fly*  
`public Animation fly( java.awt.Point p )`  
  - **Parameters**
    - \* `p` -
  - **Returns** -

---

- *fly*  
`public Animation fly( java.awt.Point p, int shadow )`  
  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* `p` -
    - \* `shadow` -
  - **Returns** -

---

- *getActorAt*  
`public Actor getActorAt( int x, int y )`  
  - **Parameters**
    - \* `x` -
    - \* `y` -
  - **Returns** -

---

- *getBackground*  
`public Color getBackground( )`  
  - **Returns** -

---

- *getFont*  
`public Font getFont( )`  
  - **Returns** -

---

- *getFontMetrics*  
`protected FontMetrics getFontMetrics( )`  
  - **Returns** -

---

- *getForeground*  
 public Color **getForeground**( )  
 – Returns -  


---
- *getHeight*  
 public int **getHeight**( )  
 – Returns -  


---
- *getLocation*  
 public Point **getLocation**( )  
 – Returns -  


---
- *getParent*  
 public ActorContainer **getParent**( )  
 – Returns -  


---
- *getRootLocation*  
 public Point **getRootLocation**( )  
 – Returns -  


---
- *getShadow*  
 public int **getShadow**( )  
 – Returns -  


---
- *getSize*  
 public Dimension **getSize**( )  
 – Returns -  


---
- *getWidth*  
 public int **getWidth**( )  
 – Returns -  


---
- *getX*  
 public int **getX**( )  
 – Returns -  


---
- *getY*  
 public int **getY**( )  
 – Returns -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – Usage  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – Parameters  
   \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – Usage  
   \* Paints the actors contained in the vector on this actor.  
 – Parameters  
   \* g -

- 
- \* actors -
- 
- *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )  
      - Parameters
        - \* g -
        - \* backImage -
        - \* xx -
        - \* yy -
        - \* w -
        - \* h -
- 
- *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )  
      - Usage
        - \* Paints the shadow of the actor. Override this in the subclasses if needed.
      - Parameters
        - \* g -
- 
- *setBackground*  
public void **setBackground**( java.awt.Color bgcolor )  
      - Parameters
        - \* bgcolor -
- 
- *setBorderWidth*  
public void **setBorderWidth**( int w )  
      - Parameters
        - \* w -
- 
- *setBounds*  
public void **setBounds**( int x, int y, int w, int h )  
      - Parameters
        - \* x -
        - \* y -
        - \* w -
        - \* h -
- 
- *setFont*  
public void **setFont**( java.awt.Font font )  
      - Parameters
        - \* font -
- 
- *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )  
      - Parameters
        - \* fgcolor -
- 
- *setInsets*  
public void **setInsets**( java.awt.Insets insets )  
      - Parameters
        - \* insets -
- 
- *setLight*  
public void **setLight**( int light )



- **Parameters**
  - \* *light* -

---

- *setLocation*  
 public void setLocation( int x, int y )
  - **Parameters**
  - \* *x* -
  - \* *y* -

---

- *setLocation*  
 public void setLocation( java.awt.Point loc )
  - **Parameters**
  - \* *loc* -

---

- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* *parent* -

---

- *setShadow*  
 public void setShadow( int s )
  - **Parameters**
  - \* *s* -

---

- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - **Parameters**
  - \* *si* -

---

- *setSize*  
 public void setSize( java.awt.Dimension d )
  - **Parameters**
  - \* *d* -

---

- *setSize*  
 public void setSize( int w, int h )
  - **Parameters**
  - \* *w* -
  - \* *h* -

### 16.2.10 CLASS CastActor

---

CastActor handles the animation of the casting of the primitive values.

#### DECLARATION

---

```
public class CastActor
extends jeliot.theater.Actor
```

CONSTRUCTORS

---

- *CastActor*

```
public CastActor( jeliot.theater.ValueActor  fromActor,
jeliot.theater.ValueActor  toActor )
```

- **Parameters**

- \* fromActor -
- \* toActor -

METHODS

---

- *cast*

```
public Animation cast( )
```

- **Returns -**

---

- *paintActor*

```
public void paintActor( java.awt.Graphics  g )
```

METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*

```
public Animation appear( java.awt.Point  loc )
```

- **Usage**

- \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.

- **Parameters**

- \* loc -

- **Returns -**

---

- *calculateSize*

```
public void calculateSize( )
```

---

- *clone*

```
public Object clone( )
```

---

- *fly*

```
public Animation fly( java.awt.Point  p )
```

- **Parameters**

- \* p -

- **Returns -**

---

- *fly*

```
public Animation fly( java.awt.Point  p, int  shadow )
```

- **Usage**

- \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.

- **Parameters**

- \* p -

- \* shadow -
  - Returns -

---
- \* *getActorAt*
  - public Actor getActorAt( int x, int y )
  - Parameters
  - \* x -
    - \* y -
  - Returns -

---
- \* *getBackground*
  - public Color getBackground( )
  - Returns -

---
- \* *getFont*
  - public Font getFont( )
  - Returns -

---
- \* *getFontMetrics*
  - protected FontMetrics getFontMetrics( )
  - Returns -

---
- \* *getForeground*
  - public Color getForeground( )
  - Returns -

---
- \* *getHeight*
  - public int getHeight( )
  - Returns -

---
- \* *getLocation*
  - public Point getLocation( )
  - Returns -

---
- \* *getParent*
  - public ActorContainer getParent( )
  - Returns -

---
- \* *getRootLocation*
  - public Point getRootLocation( )
  - Returns -

---
- \* *getShadow*
  - public int getShadow( )
  - Returns -

---
- \* *getSize*
  - public Dimension getSize( )
  - Returns -

---
- \* *getWidth*
  - public int getWidth( )
  - Returns -

---

- *getX*  
`public int getX( )`  
 – Returns -  


---
- *getY*  
`public int getY( )`  
 – Returns -  


---
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`  
 – Usage  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – Parameters  
   \* g - The Graphics object  


---
- *paintActors*  
`protected void paintActors( java.awt.Graphics g, java.util.Vector actors )`  
 – Usage  
   \* Paints the actors contained in the vector on this actor.  
 – Parameters  
   \* g -  
   \* actors -  


---
- *paintBackground*  
`protected void paintBackground( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )`  
 – Parameters  
   \* g -  
   \* backImage -  
   \* xx -  
   \* yy -  
   \* w -  
   \* h -  


---
- *paintShadow*  
`public void paintShadow( java.awt.Graphics g )`  
 – Usage  
   \* Paints the shadow of the actor. Override this in the subclasses if needed.  
 – Parameters  
   \* g -  


---
- *setBackground*  
`public void setBackground( java.awt.Color bgcolor )`  
 – Parameters  
   \* bgcolor -  


---
- *setBorderWidth*  
`public void setBorderWidth( int w )`  
 – Parameters  
   \* w -  


---
- *setBounds*  
`public void setBounds( int x, int y, int w, int h )`

- **Parameters**
  - \* **x** -
  - \* **y** -
  - \* **w** -
  - \* **h** -

---
- *setFont*  
 public void **setFont**( java.awt.Font font )
  - **Parameters**
  - \* font -

---
- *setForeground*  
 public void **setForeground**( java.awt.Color fgcolor )
  - **Parameters**
  - \* fgcolor -

---
- *setInsets*  
 public void **setInsets**( java.awt.Insets insets )
  - **Parameters**
  - \* insets -

---
- *setLight*  
 public void **setLight**( int light )
  - **Parameters**
  - \* light -

---
- *setLocation*  
 public void **setLocation**( int x, int y )
  - **Parameters**
  - \* **x** -
  - \* **y** -

---
- *setLocation*  
 public void **setLocation**( java.awt.Point loc )
  - **Parameters**
  - \* loc -

---
- *setParent*  
 public void **setParent**( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* parent -

---
- *setShadow*  
 public void **setShadow**( int s )
  - **Parameters**
  - \* s -

---
- *setShadowImage*  
 public static void **setShadowImage**( java.awt.Image si )
  - **Parameters**
  - \* si -

---
- *setSize*  
 public void **setSize**( java.awt.Dimension d )

- **Parameters**
    - \* *d* -
- 
- *setSize*

```
public void setSize( int  w, int  h )
```

    - **Parameters**
      - \* *w* -
      - \* *h* -

### 16.2.11 CLASS ComponentDragger

---

This class is not currently used by Jeliot. ComponentDragger is helping the draggin of the Theater's components.

#### DECLARATION

---

```
public class ComponentDragger
extends java.lang.Object
implements java.awt.event.MouseListener, java.awt.event.MouseMotionListener
```

#### CONSTRUCTORS

---

- *ComponentDragger*

```
public ComponentDragger( java.awt.Component  comp )
```

  - **Parameters**
    - \* *comp* -

#### METHODS

---

- *inStretchArea*

```
boolean inStretchArea( java.awt.Point  p )
```

    - **Parameters**
      - \* *p* -
    - **Returns** -
- 
- *isFree*

```
public boolean isFree( )
```

    - **Returns** -
- 
- *mouseClicked*

```
public void mouseClicked( java.awt.event.MouseEvent  e )
```
  - *mouseDragged*

```
public void mouseDragged( java.awt.event.MouseEvent  e )
```

- *mouseEntered*  
public void **mouseEntered**( java.awt.event.MouseEvent e )
- *mouseExited*  
public void **mouseExited**( java.awt.event.MouseEvent e )
- *mouseMoved*  
public void **mouseMoved**( java.awt.event.MouseEvent e )
- *mousePressed*  
public void **mousePressed**( java.awt.event.MouseEvent e )
- *mouseReleased*  
public void **mouseReleased**( java.awt.event.MouseEvent e )

### 16.2.12 CLASS ConstantBox

---

Constant box instance represents a place where all the literal constants appear during the animation.

#### DECLARATION

---

```
public class ConstantBox
extends jeliot.theater.Actor
```

#### FIELDS

---

- private Image image

—

#### CONSTRUCTORS

---

- *ConstantBox*  
public **ConstantBox**( java.awt.Image image )  
— **Parameters**  
\* image -

#### METHODS

---

- *calculateSize*  
public void **calculateSize**( )
- *paintActor*  
public void **paintActor**( java.awt.Graphics g )

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`  
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* `loc` -
  - **Returns** -

---
- *calculateSize*  
`public void calculateSize( )`  


---
- *clone*  
`public Object clone( )`  


---
- *fly*  
`public Animation fly( java.awt.Point p )`  
  - **Parameters**
    - \* `p` -
  - **Returns** -

---
- *fly*  
`public Animation fly( java.awt.Point p, int shadow )`  
  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* `p` -
    - \* `shadow` -
  - **Returns** -

---
- *getActorAt*  
`public Actor getActorAt( int x, int y )`  
  - **Parameters**
    - \* `x` -
    - \* `y` -
  - **Returns** -

---
- *getBackground*  
`public Color getBackground( )`  
  - **Returns** -

---
- *getFont*  
`public Font getFont( )`  
  - **Returns** -

---
- *getFontMetrics*  
`protected FontMetrics getFontMetrics( )`  
  - **Returns** -

---



- *getForeground*  
 public Color **getForeground**( )  
 – Returns -  


---
- *getHeight*  
 public int **getHeight**( )  
 – Returns -  


---
- *getLocation*  
 public Point **getLocation**( )  
 – Returns -  


---
- *getParent*  
 public ActorContainer **getParent**( )  
 – Returns -  


---
- *getRootLocation*  
 public Point **getRootLocation**( )  
 – Returns -  


---
- *getShadow*  
 public int **getShadow**( )  
 – Returns -  


---
- *getSize*  
 public Dimension **getSize**( )  
 – Returns -  


---
- *getWidth*  
 public int **getWidth**( )  
 – Returns -  


---
- *getX*  
 public int **getX**( )  
 – Returns -  


---
- *getY*  
 public int **getY**( )  
 – Returns -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – Usage  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – Parameters  
   \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – Usage  
   \* Paints the actors contained in the vector on this actor.  
 – Parameters  
   \* g -

- - \* actors -

---
- *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
    - **Parameters**
      - \* g -
      - \* backImage -
      - \* xx -
      - \* yy -
      - \* w -
      - \* h -

---
- *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )
    - **Usage**
      - \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - **Parameters**
      - \* g -

---
- *setBackground*  
public void **setBackground**( java.awt.Color bgcolor )
    - **Parameters**
      - \* bgcolor -

---
- *setBorderWidth*  
public void **setBorderWidth**( int w )
    - **Parameters**
      - \* w -

---
- *setBounds*  
public void **setBounds**( int x, int y, int w, int h )
    - **Parameters**
      - \* x -
      - \* y -
      - \* w -
      - \* h -

---
- *setFont*  
public void **setFont**( java.awt.Font font )
    - **Parameters**
      - \* font -

---
- *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )
    - **Parameters**
      - \* fgcolor -

---
- *setInsets*  
public void **setInsets**( java.awt.Insets insets )
    - **Parameters**
      - \* insets -

---
- *setLight*  
public void **setLight**( int light )

- **Parameters**
  - \* *light* -

---

- *setLocation*  
 public void setLocation( int x, int y )
  - **Parameters**
  - \* *x* -
  - \* *y* -

---

- *setLocation*  
 public void setLocation( java.awt.Point loc )
  - **Parameters**
  - \* *loc* -

---

- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* *parent* -

---

- *setShadow*  
 public void setShadow( int s )
  - **Parameters**
  - \* *s* -

---

- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - **Parameters**
  - \* *si* -

---

- *setSize*  
 public void setSize( java.awt.Dimension d )
  - **Parameters**
  - \* *d* -

---

- *setSize*  
 public void setSize( int w, int h )
  - **Parameters**
  - \* *w* -
  - \* *h* -

### 16.2.13 CLASS Director

---

Directs the program animation. Contains the commands to handle all the expressions and statements that are visualized.

#### DECLARATION

---

```
public class Director
extends java.lang.Object
```

FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for theater package.
- private boolean messagePause
  -
- private boolean stepByStep
  - True, if the director should stop after executing one statement.
- private Theater theatre
  - Theatre to show the animation in.
- private CodePane codePane
  - Pane showing the code. For highlighting.
- private Jeliot jeliot
  - Master Jeliot.
- private ActorFactory factory
  - Factory that produces the actors.
- private TheaterManager manager
  -
- private AnimationEngine engine
  -
- private MethodFrame currentMethodFrame
  -
- private Scratch currentScratch
  -
- private ConstantBox cbox
  -
- private ThreadController controller
  -
- private Stack scratchStack
  -
- private Stack frameStack
  -
- private boolean errorOccured
  -

- private Interpreter mCodeInterpreter
  -
- private LinesAndText lat
  -
- private int runUntilLine
  -
- private MessageFormat enterLoop
  -
- private MessageFormat continueLoop
  -
- private MessageFormat exitLoop
  -
- private MessageFormat breakLoop
  -
- private MessageFormat skipLoop
  -
- private MessageFormat arrayCreation
  -
- private MessageFormat arrayCreationDimensions
  -

## CONSTRUCTORS

---

- *Director*

```
public Director( jeliot.theater.Theater  theatre, jeliot.gui.CodePane
codePane, jeliot.Jeliot  jeliot, jeliot.theater.AnimationEngine  engine )
```

  - **Parameters**
    - \* theatre -
    - \* codePane -
    - \* jeliot -
    - \* engine -

METHODS

---

• *animateAssignment*

```
public void animateAssignment( jeliot.lang.Variable  variable,
jeliot.lang.Value  value, jeliot.lang.Value  casted, jeliot.lang.Value
returnValue, jeliot.theater.Highlight  h )
```

## – Parameters

```
* variable -
* value -
* casted -
* returnValue -
* h -
```

---

• *animateBinaryExpression*

```
public Value animateBinaryExpression( int  operator, jeliot.lang.Value
first, jeliot.lang.Value  second, jeliot.lang.Value  result, long
expressionCounter, jeliot.theater.Highlight  h )
```

## – Parameters

```
* operator -
* first -
* second -
* result -
* expressionCounter -
* h -
```

## – Returns -

• *animateCastExpression*

```
public void animateCastExpression( jeliot.lang.Value  fromValue,
jeliot.lang.Value  toValue )
```

## – Parameters

```
* fromValue -
* toValue -
```

---

• *animateInputHandling*

```
public Value animateInputHandling( java.lang.String  type,
jeliot.theater.Highlight  h )
```

## – Usage

```
* Shows an animation of the invocation of a static foreign method for handling input.
```

## – Parameters

```
* type -
* h -
```

## – Returns -

• *animateOMInvocation*

```
public Value animateOMInvocation( java.lang.String  methodCall,
jeliot.lang.Value [] args, jeliot.theater.Highlight  h )
```

## – Parameters

- \* `methodCall` -
  - \* `args` -
  - \* `h` -

– **Returns** -

---
- *animateOMInvocation*

```
public Value animateOMInvocation( java.lang.String  methodCall,
jeliot.lang.Value [] args, jeliot.theater.Highlight  h, jeliot.lang.Value
thisValue )
```

  - **Parameters**
  - \* `methodCall` -
    - \* `args` -
    - \* `h` -
    - \* `thisValue` -
  - **Returns** -

---
- *animatePostIncDec*

```
public void animatePostIncDec( int  operator, jeliot.lang.Variable  var,
jeliot.lang.Value  resVal, jeliot.theater.Highlight  h )
```

  - **Parameters**
  - \* `operator` -
    - \* `var` -
    - \* `resVal` -
    - \* `h` -

---
- *animatePreIncDec*

```
public void animatePreIncDec( int  operator, jeliot.lang.Variable  var,
jeliot.lang.Value  result, jeliot.theater.Highlight  h )
```

  - **Parameters**
  - \* `operator` -
    - \* `var` -
    - \* `result` -
    - \* `h` -

---
- *animateReturn*

```
public Actor animateReturn( jeliot.lang.Value  returnValue,
jeliot.lang.Value  casted, jeliot.theater.Highlight  h )
```

  - **Usage**
  - \* Animates a return statement
  - **Parameters**
  - \* `returnValue` -
    - \* `casted` -
    - \* `h` -
  - **Returns** -

---
- *animateSMInvocation*

```
public Value animateSMInvocation( java.lang.String  methodName,
jeliot.lang.Value [] args, jeliot.theater.Highlight  h )
```

- **Usage**
    - \* Animates the invocation of a domestic (user-defined) method.
  - **Parameters**
    - \* `methodName` -
    - \* `args` -
    - \* `h` -
  - **Returns** -

---
- *animateUnaryExpression*

```
public Value animateUnaryExpression( int  operator, jeliot.lang.Value
arg, jeliot.lang.Value  result, long  expressionCounter,
jeliot.theater.Highlight  h )
```

  - **Parameters**
    - \* `operator` -
    - \* `arg` -
    - \* `result` -
    - \* `expressionCounter` -
    - \* `h` -
  - **Returns** -

---
- *arrayCreation*

```
public void arrayCreation( int [] dims, jeliot.theater.Highlight  h )
```

  - **Parameters**
    - \* `dims` -
    - \* `h` -

---
- *beginBinaryExpression*

```
public ExpressionActor beginBinaryExpression( jeliot.lang.Value  operand,
int  operator, long  expressionReference, jeliot.theater.Highlight  h )
```

  - **Usage**
    - \* This method animates the first half of a binary expression. For example in expression `a + b` this will animate as (supposing that the value of `a` is 1): `1 + ...`

---
- *beginUnaryExpression*

```
public ExpressionActor beginUnaryExpression( int  operator,
jeliot.lang.Value  arg, long  expressionCounter, jeliot.theater.Highlight
h )
```

  - **Parameters**
    - \* `operator` -
    - \* `arg` -
    - \* `expressionCounter` -
    - \* `h` -
  - **Returns** -

---
- *branchElse*

```
public void branchElse( jeliot.lang.Value  check, jeliot.theater.Highlight
h )
```

  - **Parameters**



- \* check -
  - \* h -

---
- *branchThen*
  - public void **branchThen**( jeliot.lang.Value check, jeliot.theater.Highlight h )
  - Parameters
  - \* check -
    - \* h -

---
- *breakLoop*
  - public void **breakLoop**( java.lang.String statementName, jeliot.theater.Highlight h )
  - Parameters
  - \* statementName -
    - \* h -

---
- *breakSwitch*
  - public void **breakSwitch**( jeliot.theater.Highlight h )
  - Parameters
  - \* h -

---
- *closeExpression*
  - public void **closeExpression**( )

---
- *closeScope*
  - public void **closeScope**( )

---
- *closeScratch*
  - public void **closeScratch**( )

---
- *closeSwitch*
  - public void **closeSwitch**( jeliot.theater.Highlight h )
  - Parameters
  - \* h -

---
- *continueLoop*
  - public void **continueLoop**( java.lang.String statementName, jeliot.theater.Highlight h )
  - Parameters
  - \* statementName -
    - \* h -

---
- *continueLoop*
  - public void **continueLoop**( java.lang.String statementName, jeliot.lang.Value check, jeliot.theater.Highlight h )
  - Parameters
  - \* statementName -

\* check -  
 \* h -

---

- *declareObjectVariable*

```
public Variable declareObjectVariable( jeliot.lang.ObjectFrame of,
java.lang.String name, java.lang.String type, jeliot.theater.Highlight h
)
```

– **Parameters**

\* of -  
 \* name -  
 \* type -  
 \* h -

– **Returns** -

---

- *declareVariable*

```
public Variable declareVariable( java.lang.String name, java.lang.String
type, jeliot.theater.Highlight h )
```

– **Parameters**

\* name -  
 \* type -  
 \* h -

– **Returns** -

---

- *direct*

```
public void direct( )
```

– **Exceptions**

\* java.lang.Exception -

---

- *enterLoop*

```
public void enterLoop( java.lang.String statementName,
jeliot.theater.Highlight h )
```

– **Parameters**

\* statementName -  
 \* h -

---

- *enterLoop*

```
public void enterLoop( java.lang.String statementName, jeliot.lang.Value
check, jeliot.theater.Highlight h )
```

– **Parameters**

\* statementName -  
 \* check -  
 \* h -

---

- *exitLoop*

```
public void exitLoop( java.lang.String statementName, jeliot.lang.Value
check )
```

– **Parameters**

\* `statementName` -  
 \* `check` -

---

- *finishBinaryExpression*

```
public Value finishBinaryExpression( jeliot.lang.Value result, int
operator, jeliot.theater.ExpressionActor expr, jeliot.theater.Highlight h
)
```

– **Usage**

\* Animates the second part of a binary expression.

– **Parameters**

\* `result` -  
 \* `operator` -  
 \* `expr` -  
 \* `h` -

– **Returns** -

---

- *finishMethod*

```
public ValueActor finishMethod( jeliot.theater.Actor returnAct, long
expressionCounter )
```

– **Parameters**

\* `returnAct` -  
 \* `expressionCounter` -

– **Returns** -

---

- *finishUnaryExpression*

```
public Value finishUnaryExpression( int operator,
jeliot.theater.ExpressionActor exp, jeliot.lang.Value result, long
expressionCounter, jeliot.theater.Highlight h )
```

– **Parameters**

\* `operator` -  
 \* `exp` -  
 \* `result` -  
 \* `expressionCounter` -  
 \* `h` -

– **Returns** -

---

- *getCurrentMethodFrame*

```
public MethodFrame getCurrentMethodFrame( )
```

– **Returns** -

---

- *getCurrentScratch*

```
public Scratch getCurrentScratch( )
```

– **Returns** -

---

- *getEngine*

```
public AnimationEngine getEngine( )
```

– **Returns** -

---

- *getFactory*

```
public ActorFactory getFactory( )
```

– Returns -

---

- *getInput*

```
public Value getInput( java.lang.String  prompt,
jeliot.theater.InputValidator  validator )
```

– Parameters

\* prompt -  
\* validator -

– Returns -

---

- *getManager*

```
public TheaterManager getManager( )
```

– Returns -

---

- *getTheatre*

```
public Theater getTheatre( )
```

– Returns -

---

- *highlight*

```
public void highlight( jeliot.theater.Highlight  h )
```

– Parameters

\* h -

---

- *initiateVariableAccess*

```
public ValueActor initiateVariableAccess( jeliot.lang.Variable  var )
```

– Parameters

\* var -

– Returns -

---

- *introduceArrayLength*

```
public void introduceArrayLength( jeliot.lang.Value  length,
jeliot.lang.ArrayInstance  ai )
```

– Parameters

\* length -  
\* ai -

---

- *introduceLiteral*

```
public void introduceLiteral( jeliot.lang.Value  literal )
```

– Parameters

\* literal -

---

- *introduceReference*

```
public void introduceReference( jeliot.lang.Reference ref )
```

– **Parameters**

\* ref -

---

- *openScope*

```
public void openScope( )
```

---

- *openScratch*

```
public void openScratch( )
```

---

- *openSwitch*

```
public void openSwitch( jeliot.theater.Highlight h )
```

– **Parameters**

\* h -

---

- *output*

```
public void output( java.lang.String str )
```

– **Parameters**

\* str -

---

- *output*

```
public void output( jeliot.lang.Value val, jeliot.theater.Highlight h )
```

– **Parameters**

\* val -

\* h -

---

- *readChar*

```
public static Animator readChar( )
```

– **Returns** -

---

- *readDouble*

```
public static Animator readDouble( )
```

– **Returns** -

---

- *readInt*

```
public static Animator readInt( )
```

– **Returns** -

---

- *readString*

```
public static Animator readString( )
```

– **Returns** -

---

- *removeInstance*

```
public void removeInstance( jeliot.theater.InstanceActor actor )
```

– **Parameters**

- 
- \* actor -

---

    - *rightBinaryExpression*

public void **rightBinaryExpression**( jeliot.lang.Value operand,  
jeliot.theater.ExpressionActor expr, jeliot.theater.Highlight h )

– Parameters

\* operand -

\* expr -

\* h -

---
  - *runUntil*

public void **runUntil**( int line )

– Parameters

\* line -

---
  - *setActorFactory*

public void **setActorFactory**( jeliot.theater.ActorFactory factory )

– Parameters

\* factory -

---
  - *setController*

public void **setController**( jeliot.theater.ThreadController controller )

– Parameters

\* controller -

---
  - *setInterpreter*

public void **setInterpreter**( jeliot.mcode.Interpreter mCodeInterpreter )

– Usage

\* Used for setting the mcode interpreter

---
  - *setStep*

public void **setStep**( boolean step )

– Parameters

\* step -

---
  - *setUpMethod*

public void **setUpMethod**( java.lang.String methodName,  
jeliot.theater.Highlight h )

– Parameters

\* methodName -

\* h -

---
  - *setUpMethod*

public void **setUpMethod**( java.lang.String methodName,  
jeliot.lang.Value [] args, java.lang.String [] formalParameters,  
java.lang.String [] formalParameterTypes, jeliot.theater.Highlight h )

---

– **Usage**

- \* Called when the program enters a new user-defined method. Sets up a frame for the method.

– **Parameters**

- \* `methodName` -
  - \* `args` -
  - \* `formalParameters` -
  - \* `formalParameterTypes` -
  - \* `h` -
- 

• *setUpMethod*

```
public void setUpMethod( java.lang.String  methodName,
jeliot.lang.Value [] args, java.lang.String [] formalParameters,
java.lang.String [] formalParameterTypes, jeliot.theater.Highlight  h,
jeliot.lang.Value  thisValue )
```

– **Usage**

- \* Called when the program enters a new user-defined method. Sets up a frame for the method.

– **Parameters**

- \* `methodName` -
  - \* `args` -
  - \* `formalParameters` -
  - \* `formalParameterTypes` -
  - \* `h` -
  - \* `thisValue` -
- 

• *showArrayAccess*

```
public void showArrayAccess( jeliot.lang.VariableInArray  var,
jeliot.lang.Value [] indexVal, jeliot.lang.Value  returnVal,
jeliot.theater.Highlight  h )
```

– **Parameters**

- \* `var` -
  - \* `indexVal` -
  - \* `returnVal` -
  - \* `h` -
- 

• *showArrayCreation*

```
public void showArrayCreation( jeliot.lang.ArrayInstance  array,
jeliot.lang.Reference  ref, jeliot.lang.Value [] lenVal, long
expressionCounter, jeliot.theater.Highlight  h )
```

– **Parameters**

- \* `array` -
  - \* `ref` -
  - \* `lenVal` -
  - \* `expressionCounter` -
  - \* `h` -
- 

• *showErrorMessage*

```
public void showErrorMessage( jeliot.mcode.InterpreterError  e )
```

---

– **Parameters**

\* e -

---

• *showMessage*

private void showMessage( jeliot.theater.MessageActor message )

– **Parameters**

\* message -

---

• *showMessage*

private void showMessage( jeliot.theater.MessageActor message,  
java.awt.Point p )

– **Parameters**

\* message -

\* p -

---

• *showMessage*

private void showMessage( java.lang.String message )

– **Parameters**

\* message -

---

• *showMessage*

private void showMessage( java.lang.String [] message )

– **Parameters**

\* message -

---

• *showObjectCreation*

public void showObjectCreation( jeliot.lang.ObjectFrame of,  
jeliot.theater.Highlight h )

– **Parameters**

\* of -

\* h -

---

• *skipIf*

public void skipIf( jeliot.lang.Value check, jeliot.theater.Highlight h )

– **Parameters**

\* check -

\* h -

---

• *skipLoop*

public void skipLoop( java.lang.String statementName, jeliot.lang.Value  
check )

– **Parameters**

\* statementName -

\* check -

---



- *switchDefault*  

```
public void switchDefault( jeliot.theater.Highlight h )
```

  - **Parameters**  
 \* **h** -
- *switchSelected*  

```
public void switchSelected( jeliot.theater.Highlight h )
```

  - **Parameters**  
 \* **h** -

#### 16.2.14 CLASS Director.InputAnimator

---

##### DECLARATION

---

```
private static class Director.InputAnimator
extends jeliot.theater.Animator
```

##### FIELDS

---

- private String prompt  
 —
- private InputValidator validator  
 —

##### CONSTRUCTORS

---

- *Director.InputAnimator*  

```
private Director.InputAnimator( java.lang.String prompt,
jeliot.theater.InputValidator valid )
```

##### METHODS

---

- *animate*  

```
public void animate( jeliot.theater.Director director )
```

METHODS INHERITED FROM CLASS `jeliot.theater.Animator`

( in 16.2.7, page 1132)

- *animate*  
`public abstract void animate( jeliot.theater.Director director )`  
 – **Parameters**  
   \* `director` -  


---
- *getArgument*  
`public Value getArgument( int i )`  
 – **Parameters**  
   \* `i` -  
 – **Returns** -  


---
- *getArgumentActor*  
`protected Actor getArgumentActor( int i )`  
 – **Parameters**  
   \* `i` -  
 – **Returns** -  


---
- *getReturnValue*  
`public Value getReturnValue( )`  
 – **Returns** -  


---
- *setArgumentActors*  
`public void setArgumentActors( jeliot.theater.ValueActor [] argact )`  
 – **Parameters**  
   \* `argact` -  


---
- *setArguments*  
`public void setArguments( jeliot.lang.Value [] args )`  
 – **Parameters**  
   \* `args` -  


---
- *setReturnValue*  
`public void setReturnValue( jeliot.lang.Value v )`  
 – **Parameters**  
   \* `v` -

**16.2.15 CLASS ExpressionActor**

ExpressionActor represents a single line of a scratch the evaluation area. It Can contain any number of different actors inside that it renders.

## DECLARATION

```
public class ExpressionActor
extends jeliot.theater.Actor
implements ActorContainer
```

FIELDS

---

- private long id
  - identifis the ExpressionActors
- private Actor actors
  -
- private Point locs
  -
- private boolean bound
  -
- private int next
  -
- private int margin
  -

CONSTRUCTORS

---

- *ExpressionActor*  
 public **ExpressionActor**( int n )  
 – **Parameters**  
 \* n -  


---
- *ExpressionActor*  
 public **ExpressionActor**( int n, long i )  
 – **Parameters**  
 \* n -  
 \* i -  


---

METHODS

---

- *bind*  
 public void **bind**( jeliot.theater.Actor actor )  
 – **Parameters**  
 \* actor -  


---
- *cut*  
 public void **cut**( )  


---
- *getHeight*  
 public int **getHeight**( )  


---

- *getId*  
 public long **getId**( )  
 – Returns -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  


---
- *removeActor*  
 public void **removeActor**( jeliot.theater.Actor actor )  


---
- *reserve*  
 public Point **reserve**( jeliot.theater.Actor actor )  
 – Parameters  
   \* actor -  
 – Returns -  


---
- *setId*  
 public void **setId**( long id )  
 – Parameters  
   \* id -  


---
- *setLight*  
 public void **setLight**( int light )

---

#### METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
 public Animation **appear**( java.awt.Point loc )  
 – Usage  
   \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.  
 – Parameters  
   \* loc -  
 – Returns -  


---
- *calculateSize*  
 public void **calculateSize**( )  


---
- *clone*  
 public Object **clone**( )  


---
- *fly*  
 public Animation **fly**( java.awt.Point p )  
 – Parameters  
   \* p -  
 – Returns -  


---
- *fly*  
 public Animation **fly**( java.awt.Point p, int shadow )  
 – Usage

\* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.

– **Parameters**

\* **p** -  
\* **shadow** -

– **Returns** -

---

- *getActorAt*

public Actor **getActorAt**( int x, int y )

– **Parameters**

\* **x** -  
\* **y** -

– **Returns** -

---

- *getBackground*

public Color **getBackground**( )

– **Returns** -

---

- *getFont*

public Font **getFont**( )

– **Returns** -

---

- *getFontMetrics*

protected FontMetrics **getFontMetrics**( )

– **Returns** -

---

- *getForeground*

public Color **getForeground**( )

– **Returns** -

---

- *getHeight*

public int **getHeight**( )

– **Returns** -

---

- *getLocation*

public Point **getLocation**( )

– **Returns** -

---

- *getParent*

public ActorContainer **getParent**( )

– **Returns** -

---

- *getRootLocation*

public Point **getRootLocation**( )

– **Returns** -

---

- *getShadow*

public int **getShadow**( )

– **Returns** -

---

- *getSize*

public Dimension **getSize**( )

– **Returns** -

---

- *getWidth*  
`public int getWidth( )`  
  - **Returns -**

---
- *getX*  
`public int getX( )`  
  - **Returns -**

---
- *getY*  
`public int getY( )`  
  - **Returns -**

---
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`  
  - **Usage**  
    - \* Paints the actor on the given Graphics instance. Override this in subclasses.
  - **Parameters**  
    - \* *g* - The Graphics object

---
- *paintActors*  
`protected void paintActors( java.awt.Graphics g, java.util.Vector actors )`  
  - **Usage**  
    - \* Paints the actors contained in the vector on this actor.
  - **Parameters**  
    - \* *g* -
    - \* *actors* -

---
- *paintBackground*  
`protected void paintBackground( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )`  
  - **Parameters**  
    - \* *g* -
    - \* *backImage* -
    - \* *xx* -
    - \* *yy* -
    - \* *w* -
    - \* *h* -

---
- *paintShadow*  
`public void paintShadow( java.awt.Graphics g )`  
  - **Usage**  
    - \* Paints the shadow of the actor. Override this in the subclasses if needed.
  - **Parameters**  
    - \* *g* -

---
- *setBackground*  
`public void setBackground( java.awt.Color bgcolor )`  
  - **Parameters**  
    - \* *bgcolor* -

---
- *setBorderWidth*  
`public void setBorderWidth( int w )`  
  - **Parameters**

- 
- \* w -

---

    - *setBounds*  
public void **setBounds**( int x, int y, int w, int h )  
      - Parameters
      - \* x -
      - \* y -
      - \* w -
      - \* h -

---
  - *setFont*  
public void **setFont**( java.awt.Font font )  
    - Parameters
    - \* font -

---
  - *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )  
    - Parameters
    - \* fgcolor -

---
  - *setInsets*  
public void **setInsets**( java.awt.Insets insets )  
    - Parameters
    - \* insets -

---
  - *setLight*  
public void **setLight**( int light )  
    - Parameters
    - \* light -

---
  - *setLocation*  
public void **setLocation**( int x, int y )  
    - Parameters
    - \* x -
    - \* y -

---
  - *setLocation*  
public void **setLocation**( java.awt.Point loc )  
    - Parameters
    - \* loc -

---
  - *setParent*  
public void **setParent**( jeliot.theater.ActorContainer parent )  
    - Parameters
    - \* parent -

---
  - *setShadow*  
public void **setShadow**( int s )  
    - Parameters
    - \* s -

---
  - *setShadowImage*  
public static void **setShadowImage**( java.awt.Image si )  
    - Parameters

- \* si -
- - *setSize*  
 public void setSize( java.awt.Dimension d )  
 – Parameters  
 \* d -
- - *setSize*  
 public void setSize( int w, int h )  
 – Parameters  
 \* w -  
 \* h -

### 16.2.16 CLASS Highlight

Highlight represents a single highlight of the source code during the animation. This class contains the information about the beginning and ending line and column.

#### DECLARATION

```
public class Highlight
extends java.lang.Object
```

#### FIELDS

- private int beginLine  
 –
- private int beginColumn  
 –
- private int endLine  
 –
- private int endColumn  
 –

#### CONSTRUCTORS

- *Highlight*  


---

 protected **Highlight**( )
- *Highlight*  
 public **Highlight**( int bl, int bc, int el, int ec )  
 – Parameters  
 \* bl -  
 \* bc -  
 \* el -  
 \* ec -



METHODS

---

- *getBeginColumn*  
`public int getBeginColumn( )`  
     – **Returns** -  


---
- *getBeginLine*  
`public int getBeginLine( )`  
     – **Returns** -  


---
- *getEndColumn*  
`public int getEndColumn( )`  
     – **Returns** -  


---
- *getEndLine*  
`public int getEndLine( )`  
     – **Returns** -

**16.2.17 CLASS ImageLoader**

---

This class handles the image loading and caching for the animation.

DECLARATION

---

```
public class ImageLoader
extends java.lang.Object
```

FIELDS

---

- private static ResourceBundle bundle  
     – The resource bundle for theater package.
- private Hashtable images  
     – Maps image names to loaded images.
- private Hashtable darks  
     – Maps images to their dark counterparts.
- private Component comp  
     –
- private Toolkit toolkit  
     –
- private MediaTracker tracker  
     –

CONSTRUCTORS

---

- *ImageLoader*  
public ImageLoader( )

METHODS

---

- *darken*  
public Image darken( java.awt.Image image )  
 – **Parameters**  
   \* image -  
 – **Returns** -  


---
- *getImage*  
public Image getImage( java.lang.String name )  
 – **Parameters**  
   \* name -  
 – **Returns** -  


---
- *getLogicalImage*  
public Image getLogicalImage( java.lang.String name )  
 – **Parameters**  
   \* name -  
 – **Returns** -

**16.2.18 CLASS ImageValueActor**

---

ImageValueActor is an actor that is used when a value actor should be an image. At the moment this only happens when the value of a variable is unknown visualized as "???".

DECLARATION

---

```
public class ImageValueActor
extends jeliot.theater.ValueActor
```

FIELDS

---

- private Image image  
 –

CONSTRUCTORS

---

- *ImageValueActor*  
 public ImageValueActor( java.awt.Image image )  
 – Parameters  
 \* image -

METHODS

---

- *calcLabelPosition*  
 protected void calcLabelPosition( )
- *calculateSize*  
 public void calculateSize( )
- *paintValue*  
 public void paintValue( java.awt.Graphics g )

METHODS INHERITED FROM CLASS jeliot.theater.ValueActor

---

( in 16.2.39, page 1313)

- *calcLabelPosition*  
 protected void calcLabelPosition( )
- *calculateSize*  
 public void calculateSize( )
- *getLabel*  
 public String getLabel( )  
 – Returns -
- *getPreferredSize*  
 public Dimension getPreferredSize( )  
 – Returns -
- *paintActor*  
 public void paintActor( java.awt.Graphics g )
- *paintValue*  
 public void paintValue( java.awt.Graphics g )  
 – Parameters  
 \* g -
- *setBounds*  
 public void setBounds( int x, int y, int w, int h )
- *setLabel*  
 public void setLabel( java.lang.String valstr )  
 – Parameters  
 \* valstr -

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`  
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* `loc` -
  - **Returns** -

---

- *calculateSize*  
`public void calculateSize( )`

---

- *clone*  
`public Object clone( )`

---

- *fly*  
`public Animation fly( java.awt.Point p )`  
  - **Parameters**
    - \* `p` -
  - **Returns** -

---

- *fly*  
`public Animation fly( java.awt.Point p, int shadow )`  
  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* `p` -
    - \* `shadow` -
  - **Returns** -

---

- *getActorAt*  
`public Actor getActorAt( int x, int y )`  
  - **Parameters**
    - \* `x` -
    - \* `y` -
  - **Returns** -

---

- *getBackground*  
`public Color getBackground( )`  
  - **Returns** -

---

- *getFont*  
`public Font getFont( )`  
  - **Returns** -

---

- *getFontMetrics*  
`protected FontMetrics getFontMetrics( )`  
  - **Returns** -

---

- *getForeground*  
 public Color **getForeground**( )  
 – **Returns** -  


---
- *getHeight*  
 public int **getHeight**( )  
 – **Returns** -  


---
- *getLocation*  
 public Point **getLocation**( )  
 – **Returns** -  


---
- *getParent*  
 public ActorContainer **getParent**( )  
 – **Returns** -  


---
- *getRootLocation*  
 public Point **getRootLocation**( )  
 – **Returns** -  


---
- *getShadow*  
 public int **getShadow**( )  
 – **Returns** -  


---
- *getSize*  
 public Dimension **getSize**( )  
 – **Returns** -  


---
- *getWidth*  
 public int **getWidth**( )  
 – **Returns** -  


---
- *getX*  
 public int **getX**( )  
 – **Returns** -  


---
- *getY*  
 public int **getY**( )  
 – **Returns** -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – **Usage**  
 \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – **Parameters**  
 \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – **Usage**  
 \* Paints the actors contained in the vector on this actor.  
 – **Parameters**  
 \* g -

- - \* actors -

---
- *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
    - Parameters
      - \* g -
      - \* backImage -
      - \* xx -
      - \* yy -
      - \* w -
      - \* h -

---
- *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )
    - Usage
      - \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - Parameters
      - \* g -

---
- *setBackground*  
public void **setBackground**( java.awt.Color bgcolor )
    - Parameters
      - \* bgcolor -

---
- *setBorderWidth*  
public void **setBorderWidth**( int w )
    - Parameters
      - \* w -

---
- *setBounds*  
public void **setBounds**( int x, int y, int w, int h )
    - Parameters
      - \* x -
      - \* y -
      - \* w -
      - \* h -

---
- *setFont*  
public void **setFont**( java.awt.Font font )
    - Parameters
      - \* font -

---
- *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )
    - Parameters
      - \* fgcolor -

---
- *setInsets*  
public void **setInsets**( java.awt.Insets insets )
    - Parameters
      - \* insets -

---
- *setLight*  
public void **setLight**( int light )

- **Parameters**
  - \* *light* -

---

- *setLocation*  
 public void setLocation( int x, int y )
  - **Parameters**
  - \* *x* -
  - \* *y* -

---

- *setLocation*  
 public void setLocation( java.awt.Point loc )
  - **Parameters**
  - \* *loc* -

---

- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* *parent* -

---

- *setShadow*  
 public void setShadow( int s )
  - **Parameters**
  - \* *s* -

---

- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - **Parameters**
  - \* *si* -

---

- *setSize*  
 public void setSize( java.awt.Dimension d )
  - **Parameters**
  - \* *d* -

---

- *setSize*  
 public void setSize( int w, int h )
  - **Parameters**
  - \* *w* -
  - \* *h* -

### 16.2.19 CLASS IndexActor

---

IndexActor shows the line between the array access' indexing expression result value and the array's actual index.

#### DECLARATION

---

```
public class IndexActor
extends jeliot.theater.Actor
```

FIELDS

---

- private Actor source  
—
- private Point startPoint  
—
- private Point endPoint  
—

CONSTRUCTORS

---

- *IndexActor*  
public **IndexActor**( jeliot.theater.Actor source )  
— **Parameters**  
\* source -

METHODS

---

- *index*  
public Animation **index**( jeliot.theater.VariableInArrayActor varAct )  
— **Parameters**  
\* varAct -  
— **Returns** -  

---
- *paintActor*  
public void **paintActor**( java.awt.Graphics g )

METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
public Animation **appear**( java.awt.Point loc )  
— **Usage**  
\* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.  
— **Parameters**  
\* loc -  
— **Returns** -  

---
- *calculateSize*  
public void **calculateSize**( )
- *clone*  
public Object **clone**( )



- *fly*  
 public Animation fly( java.awt.Point p )  
 – Parameters  
   \* p -  
 – Returns -  


---
- *fly*  
 public Animation fly( java.awt.Point p, int shadow )  
 – Usage  
   \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.  
 – Parameters  
   \* p -  
   \* shadow -  
 – Returns -  


---
- *getActorAt*  
 public Actor getActorAt( int x, int y )  
 – Parameters  
   \* x -  
   \* y -  
 – Returns -  


---
- *getBackground*  
 public Color getBackground( )  
 – Returns -  


---
- *getFont*  
 public Font getFont( )  
 – Returns -  


---
- *getFontMetrics*  
 protected FontMetrics getFontMetrics( )  
 – Returns -  


---
- *getForeground*  
 public Color getForeground( )  
 – Returns -  


---
- *getHeight*  
 public int getHeight( )  
 – Returns -  


---
- *getLocation*  
 public Point getLocation( )  
 – Returns -  


---
- *getParent*  
 public ActorContainer getParent( )  
 – Returns -  


---
- *getRootLocation*  
 public Point getRootLocation( )

- **Returns** -

---

  - *getShadow*  
public int **getShadow**( )
  - **Returns** -

---

  - *getSize*  
public Dimension **getSize**( )
  - **Returns** -

---

  - *getWidth*  
public int **getWidth**( )
  - **Returns** -

---

  - *getX*  
public int **getX**( )
  - **Returns** -

---

  - *getY*  
public int **getY**( )
  - **Returns** -

---

  - *paintActor*  
public void **paintActor**( java.awt.Graphics g )
  - **Usage**
    - \* Paints the actor on the given Graphics instance. Override this in subclasses.
  - **Parameters**
    - \* g - The Graphics object

---

  - *paintActors*  
protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )
  - **Usage**
    - \* Paints the actors contained in the vector on this actor.
  - **Parameters**
    - \* g -
    - \* actors -

---

  - *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
  - **Parameters**
    - \* g -
    - \* backImage -
    - \* xx -
    - \* yy -
    - \* w -
    - \* h -

---

  - *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )
  - **Usage**
    - \* Paints the shadow of the actor. Override this in the subclasses if needed.
  - **Parameters**
    - \* g -

- 
- *setBackground*  
 public void setBackground( java.awt.Color bgcolor )  
 – Parameters  
 \* bgcolor -

---

  - *setBorderWidth*  
 public void setBorderWidth( int w )  
 – Parameters  
 \* w -

---

  - *setBounds*  
 public void setBounds( int x, int y, int w, int h )  
 – Parameters  
 \* x -  
 \* y -  
 \* w -  
 \* h -

---

  - *setFont*  
 public void setFont( java.awt.Font font )  
 – Parameters  
 \* font -

---

  - *setForeground*  
 public void setForeground( java.awt.Color fgcolor )  
 – Parameters  
 \* fgcolor -

---

  - *setInsets*  
 public void setInsets( java.awt.Insets insets )  
 – Parameters  
 \* insets -

---

  - *setLight*  
 public void setLight( int light )  
 – Parameters  
 \* light -

---

  - *setLocation*  
 public void setLocation( int x, int y )  
 – Parameters  
 \* x -  
 \* y -

---

  - *setLocation*  
 public void setLocation( java.awt.Point loc )  
 – Parameters  
 \* loc -

---

  - *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )  
 – Parameters  
 \* parent -

- 
- *setShadow*  
 public void **setShadow**( int s )  
 — **Parameters**  
 \* s -

---

  - *setShadowImage*  
 public static void **setShadowImage**( java.awt.Image si )  
 — **Parameters**  
 \* si -

---

  - *setSize*  
 public void **setSize**( java.awt.Dimension d )  
 — **Parameters**  
 \* d -

---

  - *setSize*  
 public void **setSize**( int w, int h )  
 — **Parameters**  
 \* w -  
 \* h -

### 16.2.20 CLASS InputComponent

---

InputComponent is shown when ever the executed program requests input. The InputComponent is rendered as a message label and a text field that collects the input.

#### DECLARATION

---

```
public class InputComponent
extends javax.swing.JPanel
implements java.awt.event.ActionListener
```

#### SERIALIZABLE FIELDS

---

- private InputValidator validator  
 —
- private JTextField field  
 —
- private JLabel label  
 —
- private Actor bgactor  
 —

FIELDS

---

- private InputValidator validator  
—
- private JTextField field  
—
- private JLabel label  
—
- private Actor bgactor  
—

CONSTRUCTORS

---

- *InputComponent*  
public **InputComponent**( java.lang.String **prompt**,  
jeliot.theater.InputValidator **validator** )  
— **Parameters**  
    \* **prompt** -  
    \* **validator** -

METHODS

---

- *actionPerformed*  
public void **actionPerformed**( java.awt.event.ActionEvent **evt** )
- *paintComponent*  
public void **paintComponent**( java.awt.Graphics **g** )
- *popup*  
public void **popup**( )
- *setBgactor*  
public void **setBgactor**( jeliot.theater.Actor **actor** )  
— **Parameters**  
    \* **actor** -

METHODS INHERITED FROM CLASS javax.swing.JPanel

---

- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getUI*  
public PanelUI **getUI**( )

- *getUIClassID*  
public String getUIClassID( )
- *paramString*  
protected String paramString( )
- *setUI*  
public void setUI( javax.swing.plaf.PanelUI )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS javax.swing.JComponent

---

- *\_paintImmediately*  
void \_paintImmediately( int , int , int , int )
- *<clinit>*  
static void <clinit>( )
- *addAncestorListener*  
public void addAncestorListener( javax.swing.event.AncestorListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *addVetoableChangeListener*  
public synchronized void addVetoableChangeListener( java.beans.VetoableChangeListener )
- *adjustPaintFlags*  
private void adjustPaintFlags( )
- *alwaysOnTop*  
boolean alwaysOnTop( )
- *checkIfChildObscuredBySibling*  
boolean checkIfChildObscuredBySibling( )
- *componentInputMapChanged*  
void componentInputMapChanged( javax.swing.ComponentInputMap )
- *computeVisibleRect*  
static final void computeVisibleRect( java.awt.Component , java.awt.Rectangle )
- *computeVisibleRect*  
public void computeVisibleRect( java.awt.Rectangle )
- *compWriteObjectNotify*  
void compWriteObjectNotify( )
- *contains*  
public boolean contains( int , int )
- *createToolTip*  
public JToolTip createToolTip( )
- *deregisterNextFocusableComponent*  
private void deregisterNextFocusableComponent( )

- *disable*  
public void **disable**( )
- *enable*  
public void **enable**( )
- *enableSerialization*  
void **enableSerialization**( )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , boolean , boolean )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , byte , byte )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , char , char )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , double , double )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , float , float )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , int , int )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , long , long )
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void **firePropertyChange**( java.lang.String , short , short )
- *fireVetoableChange*  
protected void **fireVetoableChange**( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext **getAccessibleContext**( )
- *getActionForKeyStroke*  
public ActionListener **getActionForKeyStroke**( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap **getActionMap**( )
- *getActionMap*  
final ActionMap **getActionMap**( boolean )
- *getAlignmentX*  
public float **getAlignmentX**( )
- *getAlignmentY*  
public float **getAlignmentY**( )
- *getAncestorListeners*  
public AncestorListener **getAncestorListeners**( )
- *getAutoscrolls*  
public boolean **getAutoscrolls**( )
- *getBorder*  
public Border **getBorder**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary **getClientProperties**( )

- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )
- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )
- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )



- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke **getRegisteredKeyStrokes**( )
- *getRootPane*  
public JRootPane **getRootPane**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean **getSuppressDropTarget**( )
- *getToolTipLocation*  
public Point **getToolTipLocation**( java.awt.event.MouseEvent )
- *getToolTipText*  
public String **getToolTipText**( )
- *getToolTipText*  
public String **getToolTipText**( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container **getTopLevelAncestor**( )
- *getTransferHandler*  
public TransferHandler **getTransferHandler**( )
- *getUIClassID*  
public String **getUIClassID**( )
- *getVerifyInputWhenFocusTarget*  
public boolean **getVerifyInputWhenFocusTarget**( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener **getVetoableChangeListeners**( )
- *getWidth*  
public int **getWidth**( )
- *getVisibleRect*  
public Rectangle **getVisibleRect**( )
- *getWriteObjCounter*  
static byte **getWriteObjCounter**( javax.swing.JComponent )
- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *grabFocus*  
public void **grabFocus**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isLightweightComponent*  
public static boolean **isLightweightComponent**( java.awt.Component )
- *isManagingFocus*  
public boolean **isManagingFocus**( )
- *isMaximumSizeSet*  
public boolean **isMaximumSizeSet**( )
- *isMinimumSizeSet*  
public boolean **isMinimumSizeSet**( )
- *isOpaque*  
public boolean **isOpaque**( )

- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )
- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )
- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )

- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent    )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent    )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object    , java.lang.Object    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int    , int    , int    , int    )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener    ,  
javax.swing.KeyStroke    , int    )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener    ,  
java.lang.String    , javax.swing.KeyStroke    , int    )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent(    )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component    )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean    )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke    )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener(  
java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener(  
java.beans.VetoableChangeListener    )
- *repaint*  
public void repaint( long    , int    , int    , int    , int    )
- *repaint*  
public void repaint( java.awt.Rectangle    )
- *requestDefaultFocus*  
public boolean requestDefaultFocus(    )
- *requestFocus*  
public void requestFocus(    )
- *requestFocus*  
public boolean requestFocus( boolean    )
- *requestFocusInWindow*  
public boolean requestFocusInWindow(    )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean    )

- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )
- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )
- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )

- *setPaintingChild*  
void **setPaintingChild**( java.awt.Component    )
- *setPreferredSize*  
public void **setPreferredSize**( java.awt.Dimension    )
- *setRequestFocusEnabled*  
public void **setRequestFocusEnabled**( boolean    )
- *setToolTipText*  
public void **setToolTipText**( java.lang.String    )
- *setTransferHandler*  
public void **setTransferHandler**( javax.swing.TransferHandler    )
- *setUI*  
protected void **setUI**( javax.swing.plaf.ComponentUI    )
- *setVerifyInputWhenFocusTarget*  
public void **setVerifyInputWhenFocusTarget**( boolean    )
- *setVisible*  
public void **setVisible**( boolean    )
- *setWriteObjCounter*  
static void **setWriteObjCounter**( javax.swing.JComponent    , byte    )
- *shouldDebugGraphics*  
int **shouldDebugGraphics**(    )
- *superProcessMouseEvent*  
void **superProcessMouseEvent**( java.awt.event.MouseEvent    )
- *unregisterKeyboardAction*  
public void **unregisterKeyboardAction**( javax.swing.KeyStroke    )
- *unregisterWithKeyboardManager*  
private void **unregisterWithKeyboardManager**(    )
- *unregisterWithKeyboardManager*  
private void **unregisterWithKeyboardManager**( javax.swing.KeyStroke    )
- *update*  
public void **update**( java.awt.Graphics    )
- *updateUI*  
public void **updateUI**(    )
- *writeObject*  
private void **writeObject**( java.io.ObjectOutputStream    )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void **<clinit>**(    )
- *add*  
public Component **add**( java.awt.Component    )
- *add*  
public Component **add**( java.awt.Component    , int    )
- *add*  
public void **add**( java.awt.Component    , java.lang.Object    )
- *add*  
public void **add**( java.awt.Component    , java.lang.Object    , int    )
- *add*  
public Component **add**( java.lang.String    , java.awt.Component    )

- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener    )
- *addImpl*  
protected void addImpl( java.awt.Component    , java.lang.Object    , int    )
- *addNotify*  
public void addNotify(    )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener    )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String    ,  
java.beans.PropertyChangeListener    )
- *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int    )
- *adjustDescendants*  
void adjustDescendants( int    )
- *adjustListeningChildren*  
void adjustListeningChildren( long    , int    )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation    )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int    )
- *checkGD*  
void checkGD( java.lang.String    )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide(    )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide(    )
- *containsFocus*  
final boolean containsFocus(    )
- *countComponents*  
public int countComponents(    )
- *countHierarchyMembers*  
int countHierarchyMembers(    )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int    , long    , boolean    )
- *createHierarchyEvents*  
int createHierarchyEvents( int    , java.awt.Component    , java.awt.Container    , long  
  , boolean    )
- *deliverEvent*  
public void deliverEvent( java.awt.Event    )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent    )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent    )
- *doLayout*  
public void doLayout(    )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent    )
- *findComponentAt*  
public Component findComponentAt( int    , int    )

- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )
- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseEventTarget*  
Component getMouseEventTarget( int , int , boolean )
- *getMouseEventTarget*  
private Component getMouseEventTarget( int , int , boolean , java.awt.Container.EventTargetFilter , boolean )

- *getMouseEventTargetImpl*  
private Component getMouseEventTargetImpl( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *insets*  
public Insets insets( )
- *invalidate*  
public void invalidate( )
- *invalidateTree*  
void invalidateTree( )
- *isAncestorOf*  
public boolean isAncestorOf( java.awt.Component )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean isFocusTraversalPolicySet( )
- *isParentOf*  
boolean isParentOf( java.awt.Component )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *minimumSize*  
public Dimension minimumSize( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintComponents*  
public void paintComponents( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )



- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent    )
- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *preProcessKeyEvent*  
void **preProcessKeyEvent**( java.awt.event.KeyEvent    )
- *print*  
public void **print**( java.awt.Graphics    )
- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )
- *removeAll*  
public void **removeAll**( )
- *removeContainerListener*  
public synchronized void **removeContainerListener**( java.awt.event.ContainerListener  
  )
- *removeNotify*  
public void **removeNotify**( )
- *setFocusCycleRoot*  
public void **setFocusCycleRoot**( boolean    )
- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int    , java.util.Set    )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy    )
- *setFont*  
public void **setFont**( java.awt.Font    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setZOrder*  
void **setZOrder**( java.awt.Component    , int    )
- *transferFocusBackward*  
public void **transferFocusBackward**( )
- *transferFocusDownCycle*  
public void **transferFocusDownCycle**( )
- *update*  
public void **update**( java.awt.Graphics    )

- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )
- *addInputMethodListener*  
public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )
- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustListeningChildrenOnParent*  
void adjustListeningChildrenOnParent( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )

- *areInputMethodsEnabled*  
boolean **areInputMethodsEnabled**( )
- *autoProcessMouseWheel*  
void **autoProcessMouseWheel**( java.awt.event.MouseWheelEvent    )
- *autoTransferFocus*  
final void **autoTransferFocus**( boolean    )
- *bounds*  
public Rectangle **bounds**( )
- *checkGD*  
void **checkGD**( java.lang.String    )
- *checkImage*  
public int **checkImage**( java.awt.Image    , java.awt.image.ImageObserver    )
- *checkImage*  
public int **checkImage**( java.awt.Image    , int    , int    , java.awt.image.ImageObserver    )
- *checkWindowClosingException*  
boolean **checkWindowClosingException**( )
- *clearCurrentFocusCycleRootOnHide*  
void **clearCurrentFocusCycleRootOnHide**( )
- *clearMostRecentFocusOwnerOnHide*  
void **clearMostRecentFocusOwnerOnHide**( )
- *coalesceEvents*  
protected AWTEvent **coalesceEvents**( java.awt.AWTEvent    , java.awt.AWTEvent    )
- *constructComponentName*  
String **constructComponentName**( )
- *contains*  
public boolean **contains**( int    , int    )
- *contains*  
public boolean **contains**( java.awt.Point    )
- *containsFocus*  
boolean **containsFocus**( )
- *countHierarchyMembers*  
int **countHierarchyMembers**( )
- *createBufferStrategy*  
void **createBufferStrategy**( int    )
- *createBufferStrategy*  
void **createBufferStrategy**( int    , java.awt.BufferCapabilities    )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int    , long    , boolean    )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int    , java.awt.Component    , java.awt.Container    , long    , boolean    )
- *createImage*  
public Image **createImage**( java.awt.image.ImageProducer    )
- *createImage*  
public Image **createImage**( int    , int    )
- *createVolatileImage*  
public VolatileImage **createVolatileImage**( int    , int    )
- *createVolatileImage*  
public VolatileImage **createVolatileImage**( int    , int    , java.awt.ImageCapabilities    )

- *deliverEvent*  
public void deliverEvent( java.awt.Event    )
- *disable*  
public void disable(    )
- *disableEvents*  
protected final void disableEvents( long    )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent    )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent    )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent    )
- *doAutoTransfer*  
private void doAutoTransfer( boolean    )
- *doLayout*  
public void doLayout(    )
- *enable*  
public void enable(    )
- *enable*  
public void enable( boolean    )
- *enableEvents*  
protected final void enableEvents( long    )
- *enableInputMethods*  
public void enableInputMethods( boolean    )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent    )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int    )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String    , boolean    , boolean    )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String    , int    , int    )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String    , java.lang.Object    , java.lang.Object    )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext(    )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent(    )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet(    )
- *getAlignmentX*  
public float getAlignmentX(    )
- *getAlignmentY*  
public float getAlignmentY(    )
- *getBackBuffer*  
Image getBackBuffer(    )
- *getBackground*  
public Color getBackground(    )
- *getBounds*  
public Rectangle getBounds(    )

- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy getBufferStrategy( )
- *getColorModel*  
public ColorModel getColorModel( )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener getComponentListeners( )
- *getComponentOrientation*  
public ComponentOrientation getComponentOrientation( )
- *getCursor*  
public Cursor getCursor( )
- *getDropTarget*  
public synchronized DropTarget getDropTarget( )
- *getFocusCycleRootAncestor*  
public Container getFocusCycleRootAncestor( )
- *getFocusListeners*  
public synchronized FocusListener getFocusListeners( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set getFocusTraversalKeys\_NoIDCheck( int )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalKeysEnabled*  
public boolean getFocusTraversalKeysEnabled( )
- *getFont\_NoClientCode*  
final Font getFont\_NoClientCode( )
- *getFont*  
public Font getFont( )
- *getFontMetrics*  
public FontMetrics getFontMetrics( java.awt.Font )
- *getForeground*  
public Color getForeground( )
- *getGraphics*  
public Graphics getGraphics( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration getGraphicsConfiguration( )
- *getHeight*  
public int getHeight( )
- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener getHierarchyBoundsListeners( )
- *getHierarchyListeners*  
public synchronized HierarchyListener getHierarchyListeners( )
- *getIgnoreRepaint*  
public boolean getIgnoreRepaint( )
- *getInputContext*  
public InputContext getInputContext( )

- *getInputMethodListeners*  
public synchronized InputMethodListener getInputMethodListeners( )
- *getInputMethodRequests*  
public InputMethodRequests getInputMethodRequests( )
- *getKeyListeners*  
public synchronized KeyListener getKeyListeners( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocale*  
public Locale getLocale( )
- *getLocation*  
public Point getLocation( )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point getLocationOnScreen\_NoTreeLock( )
- *getLocationOnScreen*  
public Point getLocationOnScreen( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseListeners*  
public synchronized MouseListener getMouseListeners( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )
- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( java.lang.String )
- *getSize*  
public Dimension getSize( )
- *getSize*  
public Dimension getSize( java.awt.Dimension )
- *getToolkit*  
public Toolkit getToolkit( )

- *getToolkitImpl*  
final Toolkit getToolkitImpl( )
- *getTreeLock*  
public final Object getTreeLock( )
- *getWidth*  
public int getWidth( )
- *getX*  
public int getX( )
- *getY*  
public int getY( )
- *gotFocus*  
public boolean gotFocus( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean handleEvent( java.awt.Event )
- *hasFocus*  
public boolean hasFocus( )
- *hide*  
public void hide( )
- *imageUpdate*  
public boolean imageUpdate( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void initializeFocusTraversalKeys( )
- *initIDs*  
private static native void initIDs( )
- *inside*  
public boolean inside( int , int )
- *invalidate*  
public void invalidate( )
- *isBackgroundSet*  
public boolean isBackgroundSet( )
- *isCursorSet*  
public boolean isCursorSet( )
- *isDisplayable*  
public boolean isDisplayable( )
- *isDoubleBuffered*  
public boolean isDoubleBuffered( )
- *isEnabled*  
public boolean isEnabled( )
- *isEnabledImpl*  
final boolean isEnabledImpl( )
- *isFocusable*  
public boolean isFocusable( )
- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )

- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )
- *minimumSize*  
public Dimension minimumSize( )
- *mouseDown*  
public boolean mouseDown( java.awt.Event , int , int )
- *mouseDrag*  
public boolean mouseDrag( java.awt.Event , int , int )
- *mouseEnter*  
public boolean mouseEnter( java.awt.Event , int , int )



- *mouseExit*  
public boolean mouseExit( java.awt.Event , int , int )
- *mouseMove*  
public boolean mouseMove( java.awt.Event , int , int )
- *mouseUp*  
public boolean mouseUp( java.awt.Event , int , int )
- *move*  
public void move( int , int )
- *nextFocus*  
public void nextFocus( )
- *nextFocusHelper*  
boolean nextFocusHelper( )
- *numListening*  
int numListening( long )
- *paint*  
public void paint( java.awt.Graphics )
- *paintAll*  
public void paintAll( java.awt.Graphics )
- *paintHeavyweightComponents*  
void paintHeavyweightComponents( java.awt.Graphics )
- *paramString*  
protected String paramString( )
- *postEvent*  
public boolean postEvent( java.awt.Event )
- *postsOldMouseEvents*  
boolean postsOldMouseEvents( )
- *preferredSize*  
public Dimension preferredSize( )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean prepareImage( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printHeavyweightComponents*  
void printHeavyweightComponents( java.awt.Graphics )
- *processComponentEvent*  
protected void processComponentEvent( java.awt.event.ComponentEvent )
- *processEvent*  
protected void processEvent( java.awt.AWTEvent )
- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent )

- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *remove*  
public synchronized void remove( java.awt.MenuComponent )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *repaint*  
public void repaint( )
- *repaint*  
public void repaint( int , int , int , int )
- *repaint*  
public void repaint( long )
- *repaint*  
public void repaint( long , int , int , int , int )

- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setGCFromPeer*  
void setGCFromPeer( )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean )
- *setLocale*  
public void setLocale( java.util.Locale )

- *setLocation*  
public void setLocation( int , int )
- *setLocation*  
public void setLocation( java.awt.Point )
- *setName*  
public void setName( java.lang.String )
- *setSize*  
public void setSize( java.awt.Dimension )
- *setSize*  
public void setSize( int , int )
- *setVisible*  
public void setVisible( boolean )
- *show*  
public void show( )
- *show*  
public void show( boolean )
- *size*  
public Dimension size( )
- *toString*  
public String toString( )
- *transferFocus*  
public void transferFocus( )
- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusUpCycle*  
public void transferFocusUpCycle( )
- *update*  
public void update( java.awt.Graphics )
- *updateCursorImmediately*  
final void updateCursorImmediately( )
- *validate*  
public void validate( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

### 16.2.21 CLASS InputValidator

---

InputValidator is the base class for all the input validators for the input component.

#### DECLARATION

---

```
public abstract class InputValidator
extends java.lang.Object
```

FIELDS

---

- private ThreadController controller

—

- private Value value

—

CONSTRUCTORS

---

- *InputValidator*  
public **InputValidator**( )

METHODS

---

- *accept*  
public void **accept**( jeliot.lang.Value value )  
— **Parameters**  
\* value -

---

- *getValue*  
public Value **getValue**( )  
— **Returns** -

---

- *isOk*  
public boolean **isOk**( )  
— **Returns** -

---

- *setController*  
public void **setController**( jeliot.theater.ThreadController controller )  
— **Parameters**  
\* controller -

---

- *validate*  
public abstract void **validate**( java.lang.String text )  
— **Parameters**  
\* text -

**16.2.22 CLASS InstanceActor**

---

InstanceActor is a base class for all the instances: ArrayActors and ObjectStage. An instance of this class should not be instantiated.

DECLARATION

---

```
public class InstanceActor
extends jeliot.theater.Actor
implements ActorContainer
```

FIELDS

---

- private Vector references

—

CONSTRUCTORS

---

- *InstanceActor*  
**protected InstanceActor( )**

METHODS

---

- *addReference*  
**public void addReference( jeliot.theater.ReferenceActor ref )**  
  
— **Parameters**  
\* ref -

---
- *getNumberOfReferences*  
**public int getNumberOfReferences( )**  
  
— **Returns -**  

---
- *removeActor*  
**public void removeActor( jeliot.theater.Actor actor )**

---
- *removeReference*  
**public void removeReference( jeliot.theater.ReferenceActor ref )**  
  
— **Parameters**  
\* ref -

---
- *setLocation*  
**public void setLocation( int x, int y )**

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`  
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* `loc` -
  - **Returns** -

---
- *calculateSize*  
`public void calculateSize( )`  


---
- *clone*  
`public Object clone( )`  


---
- *fly*  
`public Animation fly( java.awt.Point p )`  
  - **Parameters**
    - \* `p` -
  - **Returns** -

---
- *fly*  
`public Animation fly( java.awt.Point p, int shadow )`  
  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* `p` -
    - \* `shadow` -
  - **Returns** -

---
- *getActorAt*  
`public Actor getActorAt( int x, int y )`  
  - **Parameters**
    - \* `x` -
    - \* `y` -
  - **Returns** -

---
- *getBackground*  
`public Color getBackground( )`  
  - **Returns** -

---
- *getFont*  
`public Font getFont( )`  
  - **Returns** -

---
- *getFontMetrics*  
`protected FontMetrics getFontMetrics( )`  
  - **Returns** -

---

- *getForeground*  
 public Color **getForeground**( )  
 – Returns -  


---
- *getHeight*  
 public int **getHeight**( )  
 – Returns -  


---
- *getLocation*  
 public Point **getLocation**( )  
 – Returns -  


---
- *getParent*  
 public ActorContainer **getParent**( )  
 – Returns -  


---
- *getRootLocation*  
 public Point **getRootLocation**( )  
 – Returns -  


---
- *getShadow*  
 public int **getShadow**( )  
 – Returns -  


---
- *getSize*  
 public Dimension **getSize**( )  
 – Returns -  


---
- *getWidth*  
 public int **getWidth**( )  
 – Returns -  


---
- *getX*  
 public int **getX**( )  
 – Returns -  


---
- *getY*  
 public int **getY**( )  
 – Returns -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – Usage  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – Parameters  
   \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – Usage  
   \* Paints the actors contained in the vector on this actor.  
 – Parameters  
   \* g -



- - \* actors -

---
- *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
    - Parameters
      - \* g -
      - \* backImage -
      - \* xx -
      - \* yy -
      - \* w -
      - \* h -

---
- *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )
    - Usage
      - \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - Parameters
      - \* g -

---
- *setBackground*  
public void **setBackground**( java.awt.Color bgcolor )
    - Parameters
      - \* bgcolor -

---
- *setBorderWidth*  
public void **setBorderWidth**( int w )
    - Parameters
      - \* w -

---
- *setBounds*  
public void **setBounds**( int x, int y, int w, int h )
    - Parameters
      - \* x -
      - \* y -
      - \* w -
      - \* h -

---
- *setFont*  
public void **setFont**( java.awt.Font font )
    - Parameters
      - \* font -

---
- *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )
    - Parameters
      - \* fgcolor -

---
- *setInsets*  
public void **setInsets**( java.awt.Insets insets )
    - Parameters
      - \* insets -

---
- *setLight*  
public void **setLight**( int light )

- **Parameters**
  - \* *light* -

---

- *setLocation*  
 public void setLocation( int x, int y )
  - **Parameters**
  - \* *x* -
  - \* *y* -

---

- *setLocation*  
 public void setLocation( java.awt.Point loc )
  - **Parameters**
  - \* *loc* -

---

- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* *parent* -

---

- *setShadow*  
 public void setShadow( int s )
  - **Parameters**
  - \* *s* -

---

- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - **Parameters**
  - \* *si* -

---

- *setSize*  
 public void setSize( java.awt.Dimension d )
  - **Parameters**
  - \* *d* -

---

- *setSize*  
 public void setSize( int w, int h )
  - **Parameters**
  - \* *w* -
  - \* *h* -

### 16.2.23 CLASS LinesAndText

---

This actor draws the dashed lines and titles to separate explicitly the theater (animation frame) into four areas: constant area, method area, object and array area and expression evaluation area.

#### DECLARATION

---

```
public class LinesAndText
extends jeliot.theater.Actor
```

FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for theater package
- private Theater theatre
  -
- private TheaterManager manager
  -
- private String constantArea
  -
- private String methodArea
  -
- private String instanceArea
  -
- private String evaluationArea
  -
- private int constantAreaWidth
  -
- private int methodAreaWidth
  -
- private int instanceAreaWidth
  -
- private int evaluationAreaWidth
  -

CONSTRUCTORS

---

- *LinesAndText*  
public LinesAndText( )
- *LinesAndText*  
 public LinesAndText( jeliot.theater.Theater t,  
 jeliot.theater.TheaterManager tm )
  - **Parameters**
    - \* t -
    - \* tm -

METHODS

---

• *paintActor*

```
public void paintActor( java.awt.Graphics g )
```

## – Usage

\* Draws the lines separating different areas and writes texts on them.

## – See Also

\* `jeliot.theater.Actor.paintActor(java.awt.Graphics)` ( in 16.2.2, page 1113)

---

• *setManager*

```
public void setManager( jeliot.theater.TheaterManager tm )
```

## – Parameters

\* `tm` -

---

• *setTheatre*

```
public void setTheatre( jeliot.theater.Theater t )
```

## – Parameters

\* `t` -

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

• *appear*

```
public Animation appear( java.awt.Point loc )
```

## – Usage

\* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.

## – Parameters

\* `loc` -

## – Returns -

• *calculateSize*

```
public void calculateSize( )
```

---

• *clone*

```
public Object clone( )
```

---

• *fly*

```
public Animation fly( java.awt.Point p )
```

## – Parameters

\* `p` -

## – Returns -

• *fly*

```
public Animation fly( java.awt.Point p, int shadow )
```

## – Usage

\* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.

## – Parameters

- \* p -
  - \* shadow -
  - Returns -

---
- *getActorAt*
  - public Actor getActorAt( int x, int y )
  - Parameters
  - \* x -
    - \* y -
  - Returns -

---
- *getBackground*
  - public Color getBackground( )
  - Returns -

---
- *getFont*
  - public Font getFont( )
  - Returns -

---
- *getFontMetrics*
  - protected FontMetrics getFontMetrics( )
  - Returns -

---
- *getForeground*
  - public Color getForeground( )
  - Returns -

---
- *getHeight*
  - public int getHeight( )
  - Returns -

---
- *getLocation*
  - public Point getLocation( )
  - Returns -

---
- *getParent*
  - public ActorContainer getParent( )
  - Returns -

---
- *getRootLocation*
  - public Point getRootLocation( )
  - Returns -

---
- *getShadow*
  - public int getShadow( )
  - Returns -

---
- *getSize*
  - public Dimension getSize( )
  - Returns -

---
- *getWidth*
  - public int getWidth( )
  - Returns -

---

- 
- *getX*  
 public int **getX**( )  
 – **Returns** -

---

  - *getY*  
 public int **getY**( )  
 – **Returns** -

---

  - *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – **Usage**  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – **Parameters**  
   \* g - The Graphics object

---

  - *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – **Usage**  
   \* Paints the actors contained in the vector on this actor.  
 – **Parameters**  
   \* g -  
   \* actors -

---

  - *paintBackground*  
 protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )  
 – **Parameters**  
   \* g -  
   \* backImage -  
   \* xx -  
   \* yy -  
   \* w -  
   \* h -

---

  - *paintShadow*  
 public void **paintShadow**( java.awt.Graphics g )  
 – **Usage**  
   \* Paints the shadow of the actor. Override this in the subclasses if needed.  
 – **Parameters**  
   \* g -

---

  - *setBackground*  
 public void **setBackground**( java.awt.Color bgcolor )  
 – **Parameters**  
   \* bgcolor -

---

  - *setBorderWidth*  
 public void **setBorderWidth**( int w )  
 – **Parameters**  
   \* w -

---

  - *setBounds*  
 public void **setBounds**( int x, int y, int w, int h )

- **Parameters**
  - \* **x** -
  - \* **y** -
  - \* **w** -
  - \* **h** -

---
- *setFont*
  - public void setFont( java.awt.Font font )
  - **Parameters**
    - \* font -

---
- *setForeground*
  - public void setForeground( java.awt.Color fgcolor )
  - **Parameters**
    - \* fgcolor -

---
- *setInsets*
  - public void setInsets( java.awt.Insets insets )
  - **Parameters**
    - \* insets -

---
- *setLight*
  - public void setLight( int light )
  - **Parameters**
    - \* light -

---
- *setLocation*
  - public void setLocation( int x, int y )
  - **Parameters**
    - \* **x** -
    - \* **y** -

---
- *setLocation*
  - public void setLocation( java.awt.Point loc )
  - **Parameters**
    - \* loc -

---
- *setParent*
  - public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
    - \* parent -

---
- *setShadow*
  - public void setShadow( int s )
  - **Parameters**
    - \* **s** -

---
- *setShadowImage*
  - public static void setShadowImage( java.awt.Image si )
  - **Parameters**
    - \* **si** -

---
- *setSize*
  - public void setSize( java.awt.Dimension d )

---

– **Parameters**

\* **d** -

---

- *setSize*

```
public void setSize( int  w, int  h )
```

– **Parameters**

\* **w** -

\* **h** -

## 16.2.24 CLASS MessageActor

---

MessageActor shows all the textual messages to the user.

### DECLARATION

---

```
public class MessageActor
extends jeliot.theater.Actor
```

### FIELDS

---

- private int borderw

—

- private int padding

—

- private int fheight

—

- private String text

—

- private Image backgroundImage

—

### CONSTRUCTORS

---

- *MessageActor*

```
public MessageActor( )
```



METHODS

---

- *calculateSize*  
public void calculateSize( )
- *paintActor*  
public void paintActor( java.awt.Graphics g )
- *setBackground*  
public void setBackground( java.awt.Image backImage )
  - **Parameters**
    - \* backImage -
- *setText*  
public void setText( java.lang.String [] text )
  - **Parameters**
    - \* text -

METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
public Animation appear( java.awt.Point loc )
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* loc -
  - **Returns -**
- *calculateSize*  
public void calculateSize( )
- *clone*  
public Object clone( )
- *fly*  
public Animation fly( java.awt.Point p )
  - **Parameters**
    - \* p -
  - **Returns -**
- *fly*  
public Animation fly( java.awt.Point p, int shadow )
  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* p -
    - \* shadow -
  - **Returns -**

- 
- *getActorAt*  
 public Actor getActorAt( int x, int y )  
 – Parameters  
   \* x -  
   \* y -  
 – Returns -

---

  - *getBackground*  
 public Color getBackground( )  
 – Returns -

---

  - *getFont*  
 public Font getFont( )  
 – Returns -

---

  - *getFontMetrics*  
 protected FontMetrics getFontMetrics( )  
 – Returns -

---

  - *getForeground*  
 public Color getForeground( )  
 – Returns -

---

  - *getHeight*  
 public int getHeight( )  
 – Returns -

---

  - *getLocation*  
 public Point getLocation( )  
 – Returns -

---

  - *getParent*  
 public ActorContainer getParent( )  
 – Returns -

---

  - *getRootLocation*  
 public Point getRootLocation( )  
 – Returns -

---

  - *getShadow*  
 public int getShadow( )  
 – Returns -

---

  - *getSize*  
 public Dimension getSize( )  
 – Returns -

---

  - *getWidth*  
 public int getWidth( )  
 – Returns -

---

  - *getX*  
 public int getX( )

- **Returns -**

---

  - *getY*  
`public int getY( )`
    - **Returns -**

---

  - *paintActor*  
`public void paintActor( java.awt.Graphics g )`
    - **Usage**  
 \* Paints the actor on the given Graphics instance. Override this in subclasses.
    - **Parameters**  
 \* g - The Graphics object

---

  - *paintActors*  
`protected void paintActors( java.awt.Graphics g, java.util.Vector actors )`
    - **Usage**  
 \* Paints the actors contained in the vector on this actor.
    - **Parameters**  
 \* g -  
 \* actors -

---

  - *paintBackground*  
`protected void paintBackground( java.awt.Graphics g, java.awt.Image backgroundImage, int xx, int yy, int w, int h )`
    - **Parameters**  
 \* g -  
 \* backgroundImage -  
 \* xx -  
 \* yy -  
 \* w -  
 \* h -

---

  - *paintShadow*  
`public void paintShadow( java.awt.Graphics g )`
    - **Usage**  
 \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - **Parameters**  
 \* g -

---

  - *setBackground*  
`public void setBackground( java.awt.Color bgcolor )`
    - **Parameters**  
 \* bgcolor -

---

  - *setBorderWidth*  
`public void setBorderWidth( int w )`
    - **Parameters**  
 \* w -

---

  - *setBounds*  
`public void setBounds( int x, int y, int w, int h )`
    - **Parameters**  
 \* x -  
 \* y -

- \* w -
  - \* h -

---

  - *setFont*  
public void **setFont**( java.awt.Font font )  
    - Parameters
    - \* font -

---

- *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )  
  - Parameters
  - \* fgcolor -

---

- *setInsets*  
public void **setInsets**( java.awt.Insets insets )  
  - Parameters
  - \* insets -

---

- *setLight*  
public void **setLight**( int light )  
  - Parameters
  - \* light -

---

- *setLocation*  
public void **setLocation**( int x, int y )  
  - Parameters
  - \* x -
  - \* y -

---

- *setLocation*  
public void **setLocation**( java.awt.Point loc )  
  - Parameters
  - \* loc -

---

- *setParent*  
public void **setParent**( jeliot.theater.ActorContainer parent )  
  - Parameters
  - \* parent -

---

- *setShadow*  
public void **setShadow**( int s )  
  - Parameters
  - \* s -

---

- *setShadowImage*  
public static void **setShadowImage**( java.awt.Image si )  
  - Parameters
  - \* si -

---

- *setSize*  
public void **setSize**( java.awt.Dimension d )  
  - Parameters
  - \* d -

---

- *setSize*  
public void **setSize**( int w, int h )  
  - Parameters
  - \* w -
  - \* h -

### 16.2.25 CLASS MethodStage

---

MethodStage is the graphical representation of the MethodFrame. It contains the local **VariableActors** and handles the scope changes.

#### DECLARATION

---

```
public class MethodStage
extends jeliot.theater.Actor
implements ActorContainer
```

#### FIELDS

---

- private int scopeVarCount
  - Indicates how many variables is defined in this scope.
- private Stack scopes
  - Keeps track of scopes and the amount of variables in each scope.
- private Stack variables
  - Variable actors in this MethodStage.
- private String name
  - Name of the method or MethodStage.
- private int nheight
  - Height of the name.
- private int nwidth
  - Width of the name.
- private int margin
  - Number of pixels around the divider line.
- private int actorMargin
  - Number of pixels between actors.
- private int varCount
  - Maximum possible number of variables on the MethodStage at the moment.
- private int totalVarCount
  - How many variables actually is on the Methodstage at the moment.
- private int actWidth
  - Actors Width
- private int actHeight

- Actors Height
- private boolean paintVars
  - Value is true if the variables of the method are supposed to be shown. Normally, the value is true only on some animations variables are not shown. For example during appearing and disappearing this variable is set to false.
- private Actor reserved
  - Actor that is going to be added to the MethodStage but is not yet bind on it. For example an actor that is animated at the moment and then added to the MethodStage (e.g. variable appearing).
- private Point resLoc
  - The location where reserved actor is reserved.

## CONSTRUCTORS

---

- *MethodStage*  
 public MethodStage( java.lang.String name )
  - **Parameters**
    - \* name -

## METHODS

---

- *appear*  
 public Animation **appear**( java.awt.Point loc )
 

---
- *bind*  
 public void **bind**( )
 

---
- *calculateSize*  
 public void **calculateSize**( int maxActWidth, int actHeight )
  - **Parameters**
    - \* maxActWidth -
    - \* actHeight -

---
- *calculateSizeDimensions*  
 public Dimension **calculateSizeDimensions**( )
  - **Returns** -

---
- *calculateSizeDimensions*  
 public Dimension **calculateSizeDimensions**( int varCount )
  - **Parameters**
    - \* varCount -
  - **Returns** -

---

- *closeScope*  
public void closeScope( )
- *disappear*  
 public Animation disappear( )  
 – Returns -  


---
- *extend*  
 public Animation extend( )  
 – Returns -  


---
- *findVariableActor*  
 public VariableActor findVariableActor( java.lang.String name )  
 – Parameters  
     \* name -  
 – Returns -  


---
- *getActorAt*  
public Actor getActorAt( int xc, int yc )
- *openScope*  
public void openScope( )
- *paintActor*  
public void paintActor( java.awt.Graphics g )
- *removeActor*  
public void removeActor( jeliot.theater.Actor actor )
- *reserve*  
 public Point reserve( jeliot.theater.Actor actor )  
 – Parameters  
     \* actor -  
 – Returns -  


---
- *setFont*  
 public void setFont( java.awt.Font font )

---

#### METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
 public Animation appear( java.awt.Point loc )  
 – Usage  
     \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.  
 – Parameters  
     \* loc -  
 – Returns -

- 
- *calculateSize*  
`public void calculateSize( )`

---

  - *clone*  
`public Object clone( )`

---

  - *fly*  
`public Animation fly( java.awt.Point p )`
    - **Parameters**
      - \* *p* -
    - **Returns** -

---

  - *fly*  
`public Animation fly( java.awt.Point p, int shadow )`
    - **Usage**
      - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
    - **Parameters**
      - \* *p* -
      - \* *shadow* -
    - **Returns** -

---

  - *getActorAt*  
`public Actor getActorAt( int x, int y )`
    - **Parameters**
      - \* *x* -
      - \* *y* -
    - **Returns** -

---

  - *getBackground*  
`public Color getBackground( )`
    - **Returns** -

---

  - *getFont*  
`public Font getFont( )`
    - **Returns** -

---

  - *getFontMetrics*  
`protected FontMetrics getFontMetrics( )`
    - **Returns** -

---

  - *getForeground*  
`public Color getForeground( )`
    - **Returns** -

---

  - *getHeight*  
`public int getHeight( )`
    - **Returns** -

---

  - *getLocation*  
`public Point getLocation( )`
    - **Returns** -

---

  - *getParent*  
`public ActorContainer getParent( )`



- **Returns** -

---

  - *getRootLocation*  
public Point **getRootLocation**( )
  - **Returns** -

---

  - *getShadow*  
public int **getShadow**( )
  - **Returns** -

---

  - *getSize*  
public Dimension **getSize**( )
  - **Returns** -

---

  - *getWidth*  
public int **getWidth**( )
  - **Returns** -

---

  - *getX*  
public int **getX**( )
  - **Returns** -

---

  - *getY*  
public int **getY**( )
  - **Returns** -

---

- *paintActor*  
public void **paintActor**( java.awt.Graphics g )
  - **Usage**
    - \* Paints the actor on the given Graphics instance. Override this in subclasses.
  - **Parameters**
    - \* g - The Graphics object

---

- *paintActors*  
protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )
  - **Usage**
    - \* Paints the actors contained in the vector on this actor.
  - **Parameters**
    - \* g -
    - \* actors -

---

- *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
  - **Parameters**
    - \* g -
    - \* backImage -
    - \* xx -
    - \* yy -
    - \* w -
    - \* h -

---

- *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )

- **Usage**
    - \* Paints the shadow of the actor. Override this in the subclasses if needed.
  - **Parameters**
    - \* *g* -

---
- *setBackground*

```
public void setBackground( java.awt.Color bgcolor )
```

  - **Parameters**
    - \* *bgcolor* -

---
- *setBorderWidth*

```
public void setBorderWidth( int w )
```

  - **Parameters**
    - \* *w* -

---
- *setBounds*

```
public void setBounds( int x, int y, int w, int h )
```

  - **Parameters**
    - \* *x* -
    - \* *y* -
    - \* *w* -
    - \* *h* -

---
- *setFont*

```
public void setFont( java.awt.Font font )
```

  - **Parameters**
    - \* *font* -

---
- *setForeground*

```
public void setForeground( java.awt.Color fgcolor )
```

  - **Parameters**
    - \* *fgcolor* -

---
- *setInsets*

```
public void setInsets( java.awt.Insets insets )
```

  - **Parameters**
    - \* *insets* -

---
- *setLight*

```
public void setLight( int light )
```

  - **Parameters**
    - \* *light* -

---
- *setLocation*

```
public void setLocation( int x, int y )
```

  - **Parameters**
    - \* *x* -
    - \* *y* -

---
- *setLocation*

```
public void setLocation( java.awt.Point loc )
```

  - **Parameters**
    - \* *loc* -

---

- *setParent*  
 public void **setParent**( jeliot.theater.ActorContainer parent )  
 – **Parameters**  
 \* parent -  


---
- *setShadow*  
 public void **setShadow**( int s )  
 – **Parameters**  
 \* s -  


---
- *setShadowImage*  
 public static void **setShadowImage**( java.awt.Image si )  
 – **Parameters**  
 \* si -  


---
- *setSize*  
 public void **setSize**( java.awt.Dimension d )  
 – **Parameters**  
 \* d -  


---
- *setSize*  
 public void **setSize**( int w, int h )  
 – **Parameters**  
 \* w -  
 \* h -

### 16.2.26 CLASS ObjectStage

---

MethodStage is the graphical representation of the ObjectFrame. It contains the field of the object as VariableActors.

#### DECLARATION

---

```
public class ObjectStage
extends jeliot.theater.InstanceActor
```

#### FIELDS

---

- private Vector variables
  - Variable actors in this stage.
- private String name
  - Name of the stage.
- private int nheight
  - Height of the name.
- private int nwidth

- Width of the name.
- private int margin
  - Number of pixels around the divider line.
- private int actorMargin
  - Number of pixels between actors.
- private int varCount
  - Maximum possible number of variables on the stage at the moment.
- private int actWidth
  - Maximum width of the actors contained in this ObjectStage.
- private int actHeight
  - Maximum height of the actors contained in this ObjectStage.
- private boolean paintVars
  - If the variables should be drawn on top of this actor
- private Actor reserved
  - Actor that is going to be bind to this ObjectStage but is not yet bind. For example during appearing variables need their location but their are bind after they have appeared.
- private Point resLoc
  - RootLocation of the latest reserved actor.

## CONSTRUCTORS

---

- *ObjectStage*

```
public ObjectStage( java.lang.String  name, int  varCount )
```

  - **Usage**
    - \* Constructor that sets the name and variable number to the instance variables and set the insets.
  - **Parameters**
    - \* **name** - Name of the class that the instance represents.
    - \* **varCount** - Number of variables that this instance has.

## METHODS

---

- *appear*

```
public Animation appear( java.awt.Point  loc )
```
- *bind*

```
public void bind( )
```
- *calculateSize*

```
public void calculateSize( int  maxActWidth, int  actHeight )
```

- **Parameters**
    - \* `maxActWidth` -
    - \* `actHeight` -

---
- *calculateSizeDimensions*  
`public Dimension calculateSizeDimensions( )`
    - **Returns** -

---
- *calculateSizeDimensions*  
`public Dimension calculateSizeDimensions( int varCount )`
    - **Parameters**
      - \* `varCount` -
    - **Returns** -

---
- *disappear*  
`public Animation disappear( )`
    - **Returns** -

---
- *findVariableActor*  
`public VariableActor findVariableActor( java.lang.String name )`
    - **Parameters**
      - \* `name` -
    - **Returns** -

---
- *getActorAt*  
`public Actor getActorAt( int xc, int yc )`

---
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`

---
- *removeActor*  
`public void removeActor( jeliot.theater.Actor actor )`

---
- *reserve*  
`public Point reserve( jeliot.theater.Actor actor )`
    - **Parameters**
      - \* `actor` -
    - **Returns** -

---
- *setFont*  
`public void setFont( java.awt.Font font )`

METHODS INHERITED FROM CLASS `jeliot.theater.InstanceActor`

---

( in 16.2.22, page 1212)

- *addReference*  
 public void **addReference**( `jeliot.theater.ReferenceActor` `ref` )  
 – **Parameters**  
 \* `ref` -  


---
- *getNumberOfReferences*  
 public int **getNumberOfReferences**( )  
 – **Returns** -  


---
- *removeActor*  
 public void **removeActor**( `jeliot.theater.Actor` `actor` )  


---
- *removeReference*  
 public void **removeReference**( `jeliot.theater.ReferenceActor` `ref` )  
 – **Parameters**  
 \* `ref` -  


---
- *setLocation*  
 public void **setLocation**( int `x`, int `y` )

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
 public Animation **appear**( `java.awt.Point` `loc` )  
 – **Usage**  
 \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.  
 – **Parameters**  
 \* `loc` -  
 – **Returns** -  


---
- *calculateSize*  
 public void **calculateSize**( )  


---
- *clone*  
 public Object **clone**( )  


---
- *fly*  
 public Animation **fly**( `java.awt.Point` `p` )  
 – **Parameters**  
 \* `p` -  
 – **Returns** -  


---
- *fly*  
 public Animation **fly**( `java.awt.Point` `p`, int `shadow` )  
 – **Usage**  
 \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.  
 – **Parameters**

- \* p -
  - \* shadow -

– Returns -

---
- *getActorAt*

public Actor getActorAt( int x, int y )

– Parameters

  - \* x -
  - \* y -

– Returns -

---
- *getBackground*

public Color getBackground( )

– Returns -

---
- *getFont*

public Font getFont( )

– Returns -

---
- *getFontMetrics*

protected FontMetrics getFontMetrics( )

– Returns -

---
- *getForeground*

public Color getForeground( )

– Returns -

---
- *getHeight*

public int getHeight( )

– Returns -

---
- *getLocation*

public Point getLocation( )

– Returns -

---
- *getParent*

public ActorContainer getParent( )

– Returns -

---
- *getRootLocation*

public Point getRootLocation( )

– Returns -

---
- *getShadow*

public int getShadow( )

– Returns -

---
- *getSize*

public Dimension getSize( )

– Returns -

---
- *getWidth*

public int getWidth( )

– Returns -

---

- 
- *getX*  
`public int getX( )`  
    - **Returns -**
- 
- *getY*  
`public int getY( )`  
    - **Returns -**
- 
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`  
    - **Usage**  
      - \* Paints the actor on the given Graphics instance. Override this in subclasses.
    - **Parameters**  
      - \* `g` - The Graphics object
- 
- *paintActors*  
`protected void paintActors( java.awt.Graphics g, java.util.Vector actors )`  
    - **Usage**  
      - \* Paints the actors contained in the vector on this actor.
    - **Parameters**  
      - \* `g` -
      - \* `actors` -
- 
- *paintBackground*  
`protected void paintBackground( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )`  
    - **Parameters**  
      - \* `g` -
      - \* `backImage` -
      - \* `xx` -
      - \* `yy` -
      - \* `w` -
      - \* `h` -
- 
- *paintShadow*  
`public void paintShadow( java.awt.Graphics g )`  
    - **Usage**  
      - \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - **Parameters**  
      - \* `g` -
- 
- *setBackground*  
`public void setBackground( java.awt.Color bgcolor )`  
    - **Parameters**  
      - \* `bgcolor` -
- 
- *setBorderWidth*  
`public void setBorderWidth( int w )`  
    - **Parameters**  
      - \* `w` -
- 
- *setBounds*  
`public void setBounds( int x, int y, int w, int h )`



- **Parameters**
  - \* **x** -
  - \* **y** -
  - \* **w** -
  - \* **h** -

---
- *setFont*
  - public void setFont( java.awt.Font font )
  - **Parameters**
    - \* font -

---
- *setForeground*
  - public void setForeground( java.awt.Color fgcolor )
  - **Parameters**
    - \* fgcolor -

---
- *setInsets*
  - public void setInsets( java.awt.Insets insets )
  - **Parameters**
    - \* insets -

---
- *setLight*
  - public void setLight( int light )
  - **Parameters**
    - \* light -

---
- *setLocation*
  - public void setLocation( int x, int y )
  - **Parameters**
    - \* **x** -
    - \* **y** -

---
- *setLocation*
  - public void setLocation( java.awt.Point loc )
  - **Parameters**
    - \* loc -

---
- *setParent*
  - public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
    - \* parent -

---
- *setShadow*
  - public void setShadow( int s )
  - **Parameters**
    - \* **s** -

---
- *setShadowImage*
  - public static void setShadowImage( java.awt.Image si )
  - **Parameters**
    - \* **si** -

---
- *setSize*
  - public void setSize( java.awt.Dimension d )

- **Parameters**
    - \* *d* -
- 
- *setSize*

```
public void setSize( int  w, int  h )
```
  - **Parameters**
    - \* *w* -
    - \* *h* -

### 16.2.27 CLASS OMIActor

---

OMIActor represents graphically the object method invocation. The actor shows the object reference, the method name and the parameters in a similar way as Java syntax just replaces the variable references with their actual values.

#### DECLARATION

---

```
public class OMIActor
extends jeliot.theater.Actor
implements ActorContainer
```

#### CONSTRUCTORS

---

- *OMIActor*

```
public OMIActor( java.lang.String  name, int  n )
```
- **Parameters**
  - \* *name* -
  - \* *n* -

#### METHODS

---

- *bind*

```
public void bind( jeliot.theater.Actor  actor )
```
  - **Parameters**
    - \* *actor* -
- 
- *bindThisActor*

```
public void bindThisActor( )
```
  - *calculateSize*

```
public void calculateSize( )
```
  - *paintActor*

```
public void paintActor( java.awt.Graphics  g )
```
  - *paintActors*

```
public void paintActors( java.awt.Graphics  g )
```

- **Parameters**
  - \* *g* -

---
- *removeActor*  
public void removeActor( jeliot.theater.Actor actor )
- *reserve*  
 public Point reserve( jeliot.theater.Actor actor )
  - **Parameters**
  - \* actor -
  - **Returns** -

---
- *reserveThisActor*  
 public Point reserveThisActor( jeliot.theater.Actor actor )
  - **Parameters**
  - \* actor -
  - **Returns** -

---
- *setLight*  
 public void setLight( int light )

#### METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
 public Animation appear( java.awt.Point loc )
  - **Usage**
  - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
  - \* loc -
  - **Returns** -

---
- *calculateSize*  
public void calculateSize( )
- *clone*  
public Object clone( )
- *fly*  
 public Animation fly( java.awt.Point p )
  - **Parameters**
  - \* p -
  - **Returns** -

---
- *fly*  
 public Animation fly( java.awt.Point p, int shadow )
  - **Usage**
  - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**

- \* p -
  - \* shadow -

– Returns -

---
- *getActorAt*

public Actor getActorAt( int x, int y )

– Parameters

  - \* x -
  - \* y -

– Returns -

---
- *getBackground*

public Color getBackground( )

– Returns -

---
- *getFont*

public Font getFont( )

– Returns -

---
- *getFontMetrics*

protected FontMetrics getFontMetrics( )

– Returns -

---
- *getForeground*

public Color getForeground( )

– Returns -

---
- *getHeight*

public int getHeight( )

– Returns -

---
- *getLocation*

public Point getLocation( )

– Returns -

---
- *getParent*

public ActorContainer getParent( )

– Returns -

---
- *getRootLocation*

public Point getRootLocation( )

– Returns -

---
- *getShadow*

public int getShadow( )

– Returns -

---
- *getSize*

public Dimension getSize( )

– Returns -

---
- *getWidth*

public int getWidth( )

– Returns -

---

- 
- *getX*  
`public int getX( )`  
    - **Returns -**
- 
- *getY*  
`public int getY( )`  
    - **Returns -**
- 
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`  
    - **Usage**  
      - \* Paints the actor on the given Graphics instance. Override this in subclasses.
    - **Parameters**  
      - \* *g* - The Graphics object
- 
- *paintActors*  
`protected void paintActors( java.awt.Graphics g, java.util.Vector actors )`  
    - **Usage**  
      - \* Paints the actors contained in the vector on this actor.
    - **Parameters**  
      - \* *g* -
      - \* *actors* -
- 
- *paintBackground*  
`protected void paintBackground( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )`  
    - **Parameters**  
      - \* *g* -
      - \* *backImage* -
      - \* *xx* -
      - \* *yy* -
      - \* *w* -
      - \* *h* -
- 
- *paintShadow*  
`public void paintShadow( java.awt.Graphics g )`  
    - **Usage**  
      - \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - **Parameters**  
      - \* *g* -
- 
- *setBackground*  
`public void setBackground( java.awt.Color bgcolor )`  
    - **Parameters**  
      - \* *bgcolor* -
- 
- *setBorderWidth*  
`public void setBorderWidth( int w )`  
    - **Parameters**  
      - \* *w* -
- 
- *setBounds*  
`public void setBounds( int x, int y, int w, int h )`

- **Parameters**
    - \* **x** -
    - \* **y** -
    - \* **w** -
    - \* **h** -

---

- *setFont*  
 public void setFont( java.awt.Font font )
  - **Parameters**
    - \* font -

---

- *setForeground*  
 public void setForeground( java.awt.Color fgcolor )
  - **Parameters**
    - \* fgcolor -

---

- *setInsets*  
 public void setInsets( java.awt.Insets insets )
  - **Parameters**
    - \* insets -

---

- *setLight*  
 public void setLight( int light )
  - **Parameters**
    - \* light -

---

- *setLocation*  
 public void setLocation( int x, int y )
  - **Parameters**
    - \* **x** -
    - \* **y** -

---

- *setLocation*  
 public void setLocation( java.awt.Point loc )
  - **Parameters**
    - \* loc -

---

- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
    - \* parent -

---

- *setShadow*  
 public void setShadow( int s )
  - **Parameters**
    - \* **s** -

---

- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - **Parameters**
    - \* **si** -

---

- *setSize*  
 public void setSize( java.awt.Dimension d )

- **Parameters**
    - \* **d** -
- 
- *setSize*

```
public void setSize( int w, int h )
```

    - **Parameters**
      - \* **w** -
      - \* **h** -

### 16.2.28 CLASS OperatorActor

---

An instance of the OperatorActor class represents a operator in the expressions. It can be a binary or unary operator and it is shown in the ExpressionActor with the operands.

#### DECLARATION

---

```
public class OperatorActor
extends jeliot.theater.Actor
```

#### FIELDS

---

- private Image image
  -
- private Image darkImage
  -

#### CONSTRUCTORS

---

- *OperatorActor*

```
public OperatorActor( java.awt.Image image, java.awt.Image dark )
```

  - **Parameters**
    - \* **image** -
    - \* **dark** -

#### METHODS

---

- *appear*

```
public Animation appear( java.awt.Point loc )
```

  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **See Also**

\* jeliot.theater.Actor.appear(java.awt.Point) ( in 16.2.2, page 1111)

- 
- *calculateSize*  
public void **calculateSize**( )
  - *paintActor*  
public void **paintActor**( java.awt.Graphics g )

#### METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
public Animation **appear**( java.awt.Point loc )  
  
    - **Usage**  
\* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
    - **Parameters**  
\* loc -
    - **Returns** -
  - *calculateSize*  
public void **calculateSize**( )
  - *clone*  
public Object **clone**( )
  - *fly*  
public Animation **fly**( java.awt.Point p )  
  
    - **Parameters**  
\* p -
    - **Returns** -
  - *fly*  
public Animation **fly**( java.awt.Point p, int shadow )  
  
    - **Usage**  
\* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
    - **Parameters**  
\* p -  
\* shadow -
    - **Returns** -
  - *getActorAt*  
public Actor **getActorAt**( int x, int y )  
  
    - **Parameters**  
\* x -  
\* y -
    - **Returns** -
  - *getBackground*  
public Color **getBackground**( )  
  
    - **Returns** -
-



- *getFont*  
public Font **getFont**( )  
– Returns -

---
- *getFontMetrics*  
protected FontMetrics **getFontMetrics**( )  
– Returns -

---
- *getForeground*  
public Color **getForeground**( )  
– Returns -

---
- *getHeight*  
public int **getHeight**( )  
– Returns -

---
- *getLocation*  
public Point **getLocation**( )  
– Returns -

---
- *getParent*  
public ActorContainer **getParent**( )  
– Returns -

---
- *getRootLocation*  
public Point **getRootLocation**( )  
– Returns -

---
- *getShadow*  
public int **getShadow**( )  
– Returns -

---
- *getSize*  
public Dimension **getSize**( )  
– Returns -

---
- *getWidth*  
public int **getWidth**( )  
– Returns -

---
- *getX*  
public int **getX**( )  
– Returns -

---
- *getY*  
public int **getY**( )  
– Returns -

---
- *paintActor*  
public void **paintActor**( java.awt.Graphics g )  
– Usage  
\* Paints the actor on the given Graphics instance. Override this in subclasses.  
– Parameters

---

\* *g* - The Graphics object

---

- *paintActors*

protected void **paintActors**( java.awt.Graphics *g*, java.util.Vector *actors* )

- **Usage**

- \* Paints the actors contained in the vector on this actor.

- **Parameters**

- \* *g* -
    - \* *actors* -

---

- *paintBackground*

protected void **paintBackground**( java.awt.Graphics *g*, java.awt.Image *backImage*, int *xx*, int *yy*, int *w*, int *h* )

- **Parameters**

- \* *g* -
    - \* *backImage* -
    - \* *xx* -
    - \* *yy* -
    - \* *w* -
    - \* *h* -

---

- *paintShadow*

public void **paintShadow**( java.awt.Graphics *g* )

- **Usage**

- \* Paints the shadow of the actor. Override this in the subclasses if needed.

- **Parameters**

- \* *g* -

---

- *setBackground*

public void **setBackground**( java.awt.Color *bgcolor* )

- **Parameters**

- \* *bgcolor* -

---

- *setBorderWidth*

public void **setBorderWidth**( int *w* )

- **Parameters**

- \* *w* -

---

- *setBounds*

public void **setBounds**( int *x*, int *y*, int *w*, int *h* )

- **Parameters**

- \* *x* -
    - \* *y* -
    - \* *w* -
    - \* *h* -

---

- *setFont*

public void **setFont**( java.awt.Font *font* )

- **Parameters**

- \* *font* -

---

- *setForeground*

public void **setForeground**( java.awt.Color *fgcolor* )

- **Parameters**

- \* fgcolor -
 

---
- *setInsets*  
 public void **setInsets**( java.awt.Insets insets )  
 – Parameters  
 \* insets -
 

---
- *setLight*  
 public void **setLight**( int light )  
 – Parameters  
 \* light -
 

---
- *setLocation*  
 public void **setLocation**( int x, int y )  
 – Parameters  
 \* x -  
 \* y -
 

---
- *setLocation*  
 public void **setLocation**( java.awt.Point loc )  
 – Parameters  
 \* loc -
 

---
- *setParent*  
 public void **setParent**( jeliot.theater.ActorContainer parent )  
 – Parameters  
 \* parent -
 

---
- *setShadow*  
 public void **setShadow**( int s )  
 – Parameters  
 \* s -
 

---
- *setShadowImage*  
 public static void **setShadowImage**( java.awt.Image si )  
 – Parameters  
 \* si -
 

---
- *setSize*  
 public void **setSize**( java.awt.Dimension d )  
 – Parameters  
 \* d -
 

---
- *setSize*  
 public void **setSize**( int w, int h )  
 – Parameters  
 \* w -  
 \* h -
 

---

### 16.2.29 CLASS PanelActor

---

PanelActor represents the curtains of the theater and produces the opening and closing animations of the curtains.

DECLARATION

---

```
public class PanelActor
extends jeliot.theater.Actor
```

FIELDS

---

- private Image panelImage  
–
- private Image leftImage  
–
- private Image rightImage  
–
- private int lgap  
–
- private int rgap  
–
- private int gapplace  
–
- private int biw  
–
- private int lbiw  
–
- private int rbiw  
–

CONSTRUCTORS

---

- *PanelActor*  
public **PanelActor**( java.awt.Image panelImage, java.awt.Image leftImage,  
java.awt.Image rightImage, int gapplace )
  - **Parameters**
    - \* panelImage -
    - \* leftImage -
    - \* rightImage -
    - \* gapplace -

METHODS

---

- *paintActor*  
public void paintActor( java.awt.Graphics g )
- *setGap*  
public void setGap( int lgap, int rgap )
  - **Parameters**
    - \* lgap -
    - \* rgap -
- *setSize*  
public void setSize( int w, int h )
- *slide*  
public Animation slide( boolean open )
  - **Parameters**
    - \* open -
  - **Returns** -

METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
public Animation appear( java.awt.Point loc )
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* loc -
  - **Returns** -
- *calculateSize*  
public void calculateSize( )
- *clone*  
public Object clone( )
- *fly*  
public Animation fly( java.awt.Point p )
  - **Parameters**
    - \* p -
  - **Returns** -
- *fly*  
public Animation fly( java.awt.Point p, int shadow )
  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* p -

- \* shadow -
  - Returns -

---
- \* *getActorAt*
  - public Actor getActorAt( int x, int y )
  - Parameters
  - \* x -
    - \* y -
  - Returns -

---
- \* *getBackground*
  - public Color getBackground( )
  - Returns -

---
- \* *getFont*
  - public Font getFont( )
  - Returns -

---
- \* *getFontMetrics*
  - protected FontMetrics getFontMetrics( )
  - Returns -

---
- \* *getForeground*
  - public Color getForeground( )
  - Returns -

---
- \* *getHeight*
  - public int getHeight( )
  - Returns -

---
- \* *getLocation*
  - public Point getLocation( )
  - Returns -

---
- \* *getParent*
  - public ActorContainer getParent( )
  - Returns -

---
- \* *getRootLocation*
  - public Point getRootLocation( )
  - Returns -

---
- \* *getShadow*
  - public int getShadow( )
  - Returns -

---
- \* *getSize*
  - public Dimension getSize( )
  - Returns -

---
- \* *getWidth*
  - public int getWidth( )
  - Returns -

---

- *getX*  
 public int **getX**( )  
 – **Returns -**  


---
- *getY*  
 public int **getY**( )  
 – **Returns -**  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – **Usage**  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – **Parameters**  
   \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – **Usage**  
   \* Paints the actors contained in the vector on this actor.  
 – **Parameters**  
   \* g -  
   \* actors -  


---
- *paintBackground*  
 protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )  
 – **Parameters**  
   \* g -  
   \* backImage -  
   \* xx -  
   \* yy -  
   \* w -  
   \* h -  


---
- *paintShadow*  
 public void **paintShadow**( java.awt.Graphics g )  
 – **Usage**  
   \* Paints the shadow of the actor. Override this in the subclasses if needed.  
 – **Parameters**  
   \* g -  


---
- *setBackground*  
 public void **setBackground**( java.awt.Color bgcolor )  
 – **Parameters**  
   \* bgcolor -  


---
- *setBorderWidth*  
 public void **setBorderWidth**( int w )  
 – **Parameters**  
   \* w -  


---
- *setBounds*  
 public void **setBounds**( int x, int y, int w, int h )

- **Parameters**
  - \* **x** -
  - \* **y** -
  - \* **w** -
  - \* **h** -

---
- *setFont*  
 public void setFont( java.awt.Font font )
  - **Parameters**
  - \* font -

---
- *setForeground*  
 public void setForeground( java.awt.Color fgcolor )
  - **Parameters**
  - \* fgcolor -

---
- *setInsets*  
 public void setInsets( java.awt.Insets insets )
  - **Parameters**
  - \* insets -

---
- *setLight*  
 public void setLight( int light )
  - **Parameters**
  - \* light -

---
- *setLocation*  
 public void setLocation( int x, int y )
  - **Parameters**
  - \* **x** -
  - \* **y** -

---
- *setLocation*  
 public void setLocation( java.awt.Point loc )
  - **Parameters**
  - \* loc -

---
- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* parent -

---
- *setShadow*  
 public void setShadow( int s )
  - **Parameters**
  - \* **s** -

---
- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - **Parameters**
  - \* **si** -

---
- *setSize*  
 public void setSize( java.awt.Dimension d )



- **Parameters**
    - \* **d** -
- 
- *setSize*

```
public void setSize( int w, int h )
```
  - **Parameters**
    - \* **w** -
    - \* **h** -

### 16.2.30 CLASS PanelController

---

PanelController handles the curtains of the theater (PanelActor) by controlling the opening and closing of the curtains and showing the panel and background images.

#### DECLARATION

---

```
public class PanelController
extends java.lang.Object
```

#### FIELDS

---

- private static ResourceBundle bundle
  - The resource bundle for theater package
- private PanelActor panel
  -
- private Theater theatre
  -
- private AnimationEngine engine
  -
- private Image bgImage
  -
- private Image panelImage
  -
- private int openDur
  -
- private int closeDur
  -

CONSTRUCTORS

---

- *PanelController*  

```
public PanelController( jeliot.theater.Theater  theatre,
                        jeliot.theater.ImageLoader  iLoad )
```

  - **Parameters**
    - \* **theatre** -
    - \* **iLoad** -

METHODS

---

- *slide*  

```
public Thread slide( boolean  open, java.lang.Runnable  next )
```

  - **Parameters**
    - \* **open** -
    - \* **next** -
  - **Returns** -

**16.2.31 CLASS ReferenceActor**

---

ReferenceActor shows the reference to some InstanceActor (e.g. ArrayActor or ObjectStage). They can be assigned to the ReferenceVariableActor instances or any other instance that is derived from the ReferenceVariableActor.

DECLARATION

---

```
public class ReferenceActor
extends jeliot.theater.ValueActor
```

FIELDS

---

- private InstanceActor instance
  -
- private VariableActor variable
  -
- private static int refWidth
  - Reference width is the width of the rectangle in the variable end of the reference.
- private static int refLen
  - Length of the null reference line and also the base for the first part of the reference when not null.

- private int refWidthRandom
  - Used to make the first part of the reference line different in length.
- private boolean instVarConnect
  -
- private Point bend
  -
- private Point arrowhead
  -
- private Polygon arrowheadPolygon1
  -
- private Polygon arrowheadPolygon2
  -

## CONSTRUCTORS

---

- *ReferenceActor*  
**public ReferenceActor( )**


---
- *ReferenceActor*  
**public ReferenceActor( jeliot.theater.InstanceActor inst )**
  - **Parameters**
    - \* inst -

---
- *ReferenceActor*  
**public ReferenceActor( jeliot.theater.InstanceActor inst, boolean instVarConnect )**
  - **Parameters**
    - \* inst -
    - \* instVarConnect -

---
- *ReferenceActor*  
**public ReferenceActor( jeliot.theater.InstanceActor inst, jeliot.theater.VariableActor var )**
  - **Parameters**
    - \* inst -
    - \* var -

---
- *ReferenceActor*  
**public ReferenceActor( jeliot.theater.InstanceActor inst, jeliot.theater.VariableActor var, boolean instVarConnect )**
  - **Parameters**
    - \* inst -
    - \* var -
    - \* instVarConnect -

---

METHODS

---

- *calculateArrowhead*

`public void calculateArrowhead( int dir )`

- **Usage**

- \* dir is 1 up, 2 right, 3 down and 4 left.

- **Parameters**

- \* dir -

---

- *calculateBends*

`public void calculateBends( )`

---

- *calculateSize*

`public void calculateSize( )`

---

- *getInstanceActor*

`public InstanceActor getInstanceActor( )`

- **Returns -**

---

- *getPreferredSize*

`public Dimension getPreferredSize( )`

---

- *getReferenceWidth*

`public int getReferenceWidth( )`

- **Returns -**

---

- *getVariableActor*

`public VariableActor getVariableActor( )`

- **Returns -**

---

- *paintActor*

`public void paintActor( java.awt.Graphics g )`

---

- *setInstanceActor*

`public void setInstanceActor( jeliot.theater.InstanceActor inst )`

- **Parameters**

- \* inst -

---

- *setVariableActor*

`public void setVariableActor( jeliot.theater.VariableActor var )`

- **Parameters**

- \* var -

METHODS INHERITED FROM CLASS `jeliot.theater.ValueActor`

---

( in 16.2.39, page 1313)

- *calcLabelPosition*  
protected void calcLabelPosition( )
- *calculateSize*  
public void calculateSize( )
- *getLabel*  
public String getLabel( )  
  
– Returns -
- *getPreferredSize*  
public Dimension getPreferredSize( )  
  
– Returns -
- *paintActor*  
public void paintActor( java.awt.Graphics g )
- *paintValue*  
public void paintValue( java.awt.Graphics g )  
  
– Parameters  
\* g -
- *setBounds*  
public void setBounds( int x, int y, int w, int h )
- *setLabel*  
public void setLabel( java.lang.String valstr )  
  
– Parameters  
\* valstr -

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
public Animation appear( java.awt.Point loc )  
  
– Usage  
\* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
- Parameters  
\* loc -
- Returns -
- *calculateSize*  
public void calculateSize( )
- *clone*  
public Object clone( )
- *fly*  
public Animation fly( java.awt.Point p )  
  
– Parameters  
\* p -
- Returns -

- 
- *fly*  
 public Animation fly( java.awt.Point p, int shadow )  
 – Usage  
   \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.  
 – Parameters  
   \* p -  
   \* shadow -  
 – Returns -

---

  - *getActorAt*  
 public Actor getActorAt( int x, int y )  
 – Parameters  
   \* x -  
   \* y -  
 – Returns -

---

  - *getBackground*  
 public Color getBackground( )  
 – Returns -

---

  - *getFont*  
 public Font getFont( )  
 – Returns -

---

  - *getFontMetrics*  
 protected FontMetrics getFontMetrics( )  
 – Returns -

---

  - *getForeground*  
 public Color getForeground( )  
 – Returns -

---

  - *getHeight*  
 public int getHeight( )  
 – Returns -

---

  - *getLocation*  
 public Point getLocation( )  
 – Returns -

---

  - *getParent*  
 public ActorContainer getParent( )  
 – Returns -

---

  - *getRootLocation*  
 public Point getRootLocation( )  
 – Returns -

---

  - *getShadow*  
 public int getShadow( )  
 – Returns -

---

- *getSize*  
 public Dimension **getSize**( )  
 – Returns -  


---
- *getWidth*  
 public int **getWidth**( )  
 – Returns -  


---
- *getX*  
 public int **getX**( )  
 – Returns -  


---
- *getY*  
 public int **getY**( )  
 – Returns -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – Usage  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – Parameters  
   \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – Usage  
   \* Paints the actors contained in the vector on this actor.  
 – Parameters  
   \* g -  
   \* actors -  


---
- *paintBackground*  
 protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )  
 – Parameters  
   \* g -  
   \* backImage -  
   \* xx -  
   \* yy -  
   \* w -  
   \* h -  


---
- *paintShadow*  
 public void **paintShadow**( java.awt.Graphics g )  
 – Usage  
   \* Paints the shadow of the actor. Override this in the subclasses if needed.  
 – Parameters  
   \* g -  


---
- *setBackground*  
 public void **setBackground**( java.awt.Color bgcolor )  
 – Parameters  
   \* bgcolor -  


---

- *setBorderWidth*  
 public void **setBorderWidth**( int w )  
 – **Parameters**  
 \* w -  


---
- *setBounds*  
 public void **setBounds**( int x, int y, int w, int h )  
 – **Parameters**  
 \* x -  
 \* y -  
 \* w -  
 \* h -  


---
- *setFont*  
 public void **setFont**( java.awt.Font font )  
 – **Parameters**  
 \* font -  


---
- *setForeground*  
 public void **setForeground**( java.awt.Color fgcolor )  
 – **Parameters**  
 \* fgcolor -  


---
- *setInsets*  
 public void **setInsets**( java.awt.Insets insets )  
 – **Parameters**  
 \* insets -  


---
- *setLight*  
 public void **setLight**( int light )  
 – **Parameters**  
 \* light -  


---
- *setLocation*  
 public void **setLocation**( int x, int y )  
 – **Parameters**  
 \* x -  
 \* y -  


---
- *setLocation*  
 public void **setLocation**( java.awt.Point loc )  
 – **Parameters**  
 \* loc -  


---
- *setParent*  
 public void **setParent**( jeliot.theater.ActorContainer parent )  
 – **Parameters**  
 \* parent -  


---
- *setShadow*  
 public void **setShadow**( int s )  
 – **Parameters**  
 \* s -  


---



- *setShadowImage*  
 public static void **setShadowImage**( java.awt.Image si )  
 — **Parameters**  
 \* si -  


---
- *setSize*  
 public void **setSize**( java.awt.Dimension d )  
 — **Parameters**  
 \* d -  


---
- *setSize*  
 public void **setSize**( int w, int h )  
 — **Parameters**  
 \* w -  
 \* h -

### 16.2.32 CLASS ReferenceVariableActor

---

ReferenceVariableActor represents graphically the variables of the reference type. It can bind ReferenceActor instances and render them.

#### DECLARATION

---

```
public class ReferenceVariableActor
extends jeliot.theater.VariableActor
```

#### FIELDS

---

- private int refWidth  
 —
- private int refLen  
 —
- private ReferenceActor refActor  
 —
- private ReferenceActor reservedRefActor  
 —

#### CONSTRUCTORS

---

- *ReferenceVariableActor*  
 public **ReferenceVariableActor**( )

## METHODS

- 
- *bind*  
public void bind( )
  - *calculateSize*  
public void calculateSize( )
  - *getValue*  
public ValueActor getValue( )
  - *paintActor*  
public void paintActor( java.awt.Graphics g )
  - *reserve*  
public Point reserve( jeliot.theater.ReferenceActor actor )
    - **Parameters**
    - \* actor -
    - **Returns** -
  - *reserve*  
public Point reserve( jeliot.theater.ValueActor actor )
  - *setBounds*  
public void setBounds( int x, int y, int w, int h )
  - *setReference*  
public void setReference( jeliot.theater.ReferenceActor refActor )
    - **Parameters**
    - \* refActor -
  - *setValue*  
public void setValue( jeliot.theater.ReferenceActor actor )
    - **Parameters**
    - \* actor -
  - *theatreResized*  
public void theatreResized( )

## METHODS INHERITED FROM CLASS jeliot.theater.VariableActor

---

( in 16.2.40, page 1318)

- *bind*  
public void bind( )
- *calcLabelPosition*  
protected void calcLabelPosition( )
- *calculateSize*  
public void calculateSize( )
- *getName*  
public String getName( )

- **Returns** -

---

  - *getValue*  
`public ValueActor getValue( )`
    - **Returns** -

---

  - *paintActor*  
`public void paintActor( java.awt.Graphics g )`
  - *removeActor*  
`public void removeActor( jeliot.theater.Actor actor )`
  - *reserve*  
`public Point reserve( jeliot.theater.ValueActor actor )`
    - **Parameters**
      - \* actor -
    - **Returns** -

---

  - *setFont*  
`public void setFont( java.awt.Font font )`
  - *setName*  
`public void setName( java.lang.String name )`
    - **Parameters**
      - \* name -

---

  - *setSize*  
`public void setSize( int w, int h )`
  - *setValue*  
`public void setValue( jeliot.theater.ValueActor actor )`
    - **Parameters**
      - \* actor -

---

  - *setValueColor*  
`public void setValueColor( java.awt.Color valuec )`
    - **Parameters**
      - \* valuec -

---

  - *setValueDimension*  
`public void setValueDimension( int w, int h )`
    - **Parameters**
      - \* w -
      - \* h -

#### METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* loc -

– **Returns** -

- *calculateSize*

---

```
public void calculateSize( )
```

---

- *clone*

---

```
public Object clone( )
```

---

- *fly*

```
public Animation fly( java.awt.Point p )
```

– **Parameters**

\* *p* -

– **Returns** -

- 
- *fly*

```
public Animation fly( java.awt.Point p, int shadow )
```

– **Usage**

\* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.

– **Parameters**

\* *p* -

\* *shadow* -

– **Returns** -

- 
- *getActorAt*

```
public Actor getActorAt( int x, int y )
```

– **Parameters**

\* *x* -

\* *y* -

– **Returns** -

- 
- *getBackground*

```
public Color getBackground( )
```

– **Returns** -

- 
- *getFont*

```
public Font getFont( )
```

– **Returns** -

- 
- *getFontMetrics*

```
protected FontMetrics getFontMetrics( )
```

– **Returns** -

- 
- *getForeground*

```
public Color getForeground( )
```

– **Returns** -

- 
- *getHeight*

```
public int getHeight( )
```

– **Returns** -

- 
- *getLocation*

```
public Point getLocation( )
```

– **Returns** -

---

- *getParent*  
`public ActorContainer getParent( )`  


---

  - **Returns -**
- *getRootLocation*  
`public Point getRootLocation( )`  


---

  - **Returns -**
- *getShadow*  
`public int getShadow( )`  


---

  - **Returns -**
- *getSize*  
`public Dimension getSize( )`  


---

  - **Returns -**
- *getWidth*  
`public int getWidth( )`  


---

  - **Returns -**
- *getX*  
`public int getX( )`  


---

  - **Returns -**
- *getY*  
`public int getY( )`  


---

  - **Returns -**
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`  


---

  - **Usage**  
    - \* Paints the actor on the given Graphics instance. Override this in subclasses.
  - **Parameters**  
    - \* `g` - The Graphics object
- *paintActors*  
`protected void paintActors( java.awt.Graphics g, java.util.Vector actors )`  


---

  - **Usage**  
    - \* Paints the actors contained in the vector on this actor.
  - **Parameters**  
    - \* `g` -
    - \* `actors` -
- *paintBackground*  
`protected void paintBackground( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )`  


---

  - **Parameters**  
    - \* `g` -
    - \* `backImage` -
    - \* `xx` -
    - \* `yy` -
    - \* `w` -
    - \* `h` -

- *paintShadow*  
 public void **paintShadow**( java.awt.Graphics g )  
 – **Usage**  
   \* Paints the shadow of the actor. Override this in the subclasses if needed.  
 – **Parameters**  
   \* g -  


---
- *setBackground*  
 public void **setBackground**( java.awt.Color bgcolor )  
 – **Parameters**  
   \* bgcolor -  


---
- *setBorderWidth*  
 public void **setBorderWidth**( int w )  
 – **Parameters**  
   \* w -  


---
- *setBounds*  
 public void **setBounds**( int x, int y, int w, int h )  
 – **Parameters**  
   \* x -  
   \* y -  
   \* w -  
   \* h -  


---
- *setFont*  
 public void **setFont**( java.awt.Font font )  
 – **Parameters**  
   \* font -  


---
- *setForeground*  
 public void **setForeground**( java.awt.Color fgcolor )  
 – **Parameters**  
   \* fgcolor -  


---
- *setInsets*  
 public void **setInsets**( java.awt.Insets insets )  
 – **Parameters**  
   \* insets -  


---
- *setLight*  
 public void **setLight**( int light )  
 – **Parameters**  
   \* light -  


---
- *setLocation*  
 public void **setLocation**( int x, int y )  
 – **Parameters**  
   \* x -  
   \* y -  


---
- *setLocation*  
 public void **setLocation**( java.awt.Point loc )

- **Parameters**
  - \* loc -

---

- *setParent*  
 public void **setParent**( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* parent -

---

- *setShadow*  
 public void **setShadow**( int s )
  - **Parameters**
  - \* s -

---

- *setShadowImage*  
 public static void **setShadowImage**( java.awt.Image si )
  - **Parameters**
  - \* si -

---

- *setSize*  
 public void **setSize**( java.awt.Dimension d )
  - **Parameters**
  - \* d -

---

- *setSize*  
 public void **setSize**( int w, int h )
  - **Parameters**
  - \* w -
  - \* h -

### 16.2.33 CLASS Scratch

---

Scratch controls the expression evaluation area. It allocates the space for each ExpressionEvaluationActor and possible other Actors that area there temporarily.

#### DECLARATION

---

```
public class Scratch
extends jeliot.theater.Actor
implements ActorContainer
```

#### CONSTRUCTORS

---

- *Scratch*  
 public **Scratch**( )

METHODS

---

- *accommodate*  
public Point accommodate( jeliot.theater.Actor actor )
  - **Parameters**
    - \* actor -
  - **Returns** -

---
- *clean*  
public void clean( )

---
- *findActor*  
public ExpressionActor findActor( long number )
  - **Parameters**
    - \* number -
  - **Returns** -

---
- *fly*  
public Animation fly( java.awt.Point p )

---
- *getExpression*  
public ExpressionActor getExpression( int n, long id )
  - **Usage**
    - \* Second parameter added for Jeliot 3 to identify the expressions.
  - **Parameters**
    - \* n -
    - \* id -
  - **Returns** -

---
- *getSpot*  
public Point getSpot( )
  - **Returns** -

---
- *memorizeLocation*  
public void memorizeLocation( )

---
- *paintActor*  
public void paintActor( java.awt.Graphics g )

---
- *recallLocation*  
public Point recallLocation( )
  - **Returns** -

---
- *registerCrap*  
public void registerCrap( jeliot.theater.Actor actor )
  - **Parameters**
    - \* actor -



---

- *registerCrapRemover*

**public void registerCrapRemover( java.lang.Runnable remover )**

– **Parameters**

\* *remover* -

---

- *removeActor*

**public void removeActor( jeliot.theater.Actor actor )**

---

- *removeCrap*

**public void removeCrap( )**

---

- *removeCrap*

**public void removeCrap( jeliot.theater.Actor actor )**

– **Parameters**

\* *actor* -

#### METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*

**public Animation appear( java.awt.Point loc )**

– **Usage**

\* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.

– **Parameters**

\* *loc* -

– **Returns** -

---

- *calculateSize*

**public void calculateSize( )**

---

- *clone*

**public Object clone( )**

---

- *fly*

**public Animation fly( java.awt.Point p )**

– **Parameters**

\* *p* -

– **Returns** -

---

- *fly*

**public Animation fly( java.awt.Point p, int shadow )**

– **Usage**

\* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.

– **Parameters**

\* *p* -

\* *shadow* -

– **Returns** -

---

- *getActorAt*  
 public Actor **getActorAt**( int x, int y )  
 – Parameters  
   \* x -  
   \* y -  
 – Returns -  


---
- *getBackground*  
 public Color **getBackground**( )  
 – Returns -  


---
- *getFont*  
 public Font **getFont**( )  
 – Returns -  


---
- *getFontMetrics*  
 protected FontMetrics **getFontMetrics**( )  
 – Returns -  


---
- *getForeground*  
 public Color **getForeground**( )  
 – Returns -  


---
- *getHeight*  
 public int **getHeight**( )  
 – Returns -  


---
- *getLocation*  
 public Point **getLocation**( )  
 – Returns -  


---
- *getParent*  
 public ActorContainer **getParent**( )  
 – Returns -  


---
- *getRootLocation*  
 public Point **getRootLocation**( )  
 – Returns -  


---
- *getShadow*  
 public int **getShadow**( )  
 – Returns -  


---
- *getSize*  
 public Dimension **getSize**( )  
 – Returns -  


---
- *getWidth*  
 public int **getWidth**( )  
 – Returns -  


---
- *getX*  
 public int **getX**( )  
 – Returns -

- 
- *getY*  
`public int getY( )`  
    - **Returns -**
- 
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`  
    - **Usage**  
      - \* Paints the actor on the given Graphics instance. Override this in subclasses.
    - **Parameters**  
      - \* *g* - The Graphics object
- 
- *paintActors*  
`protected void paintActors( java.awt.Graphics g, java.util.Vector actors )`  
    - **Usage**  
      - \* Paints the actors contained in the vector on this actor.
    - **Parameters**  
      - \* *g* -
      - \* *actors* -
- 
- *paintBackground*  
`protected void paintBackground( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )`  
    - **Parameters**  
      - \* *g* -
      - \* *backImage* -
      - \* *xx* -
      - \* *yy* -
      - \* *w* -
      - \* *h* -
- 
- *paintShadow*  
`public void paintShadow( java.awt.Graphics g )`  
    - **Usage**  
      - \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - **Parameters**  
      - \* *g* -
- 
- *setBackground*  
`public void setBackground( java.awt.Color bgcolor )`  
    - **Parameters**  
      - \* *bgcolor* -
- 
- *setBorderWidth*  
`public void setBorderWidth( int w )`  
    - **Parameters**  
      - \* *w* -
- 
- *setBounds*  
`public void setBounds( int x, int y, int w, int h )`  
    - **Parameters**  
      - \* *x* -
      - \* *y* -
      - \* *w* -
-

- \* h -
 

---
- *setFont*  
 public void setFont( java.awt.Font font )  
 – Parameters  
 \* font -
 

---
- *setForeground*  
 public void setForeground( java.awt.Color fgcolor )  
 – Parameters  
 \* fgcolor -
 

---
- *setInsets*  
 public void setInsets( java.awt.Insets insets )  
 – Parameters  
 \* insets -
 

---
- *setLight*  
 public void setLight( int light )  
 – Parameters  
 \* light -
 

---
- *setLocation*  
 public void setLocation( int x, int y )  
 – Parameters  
 \* x -  
 \* y -
 

---
- *setLocation*  
 public void setLocation( java.awt.Point loc )  
 – Parameters  
 \* loc -
 

---
- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )  
 – Parameters  
 \* parent -
 

---
- *setShadow*  
 public void setShadow( int s )  
 – Parameters  
 \* s -
 

---
- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )  
 – Parameters  
 \* si -
 

---
- *setSize*  
 public void setSize( java.awt.Dimension d )  
 – Parameters  
 \* d -
 

---
- *setSize*  
 public void setSize( int w, int h )  
 – Parameters  
 \* w -  
 \* h -
 

---

### 16.2.34 CLASS SMIActor

---

SMIActor represents graphically the static method invocation. The actor shows the method name and the parameters in a similar way as Java syntax just replaces the variable references with their actual values.

#### DECLARATION

---

```
public class SMIActor
extends jeliot.theater.Actor
implements ActorContainer
```

#### CONSTRUCTORS

---

- *SMIActor*  

```
public SMIActor( java.lang.String  name, int  n )
```

  - **Parameters**
    - \* **name** -
    - \* **n** -

#### METHODS

---

- *bind*  

```
public void bind( jeliot.theater.Actor  actor )
```

  - **Parameters**
    - \* **actor** -
- *calculateSize*  

```
public void calculateSize( )
```
- *paintActor*  

```
public void paintActor( java.awt.Graphics  g )
```
- *paintActors*  

```
public void paintActors( java.awt.Graphics  g )
```

  - **Parameters**
    - \* **g** -
- *removeActor*  

```
public void removeActor( jeliot.theater.Actor  actor )
```
- *reserve*  

```
public Point reserve( jeliot.theater.Actor  actor )
```

  - **Parameters**
    - \* **actor** -
  - **Returns** -
- *setLight*  

```
public void setLight( int  light )
```

METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`  
  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* `loc` -
  - **Returns** -

---

- *calculateSize*  
`public void calculateSize( )`

---

- *clone*  
`public Object clone( )`

---

- *fly*  
`public Animation fly( java.awt.Point p )`  
  - **Parameters**
    - \* `p` -
  - **Returns** -

---

- *fly*  
`public Animation fly( java.awt.Point p, int shadow )`  
  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* `p` -
    - \* `shadow` -
  - **Returns** -

---

- *getActorAt*  
`public Actor getActorAt( int x, int y )`  
  - **Parameters**
    - \* `x` -
    - \* `y` -
  - **Returns** -

---

- *getBackground*  
`public Color getBackground( )`  
  - **Returns** -

---

- *getFont*  
`public Font getFont( )`  
  - **Returns** -

---

- *getFontMetrics*  
`protected FontMetrics getFontMetrics( )`  
  - **Returns** -

---

- *getForeground*  
 public Color **getForeground**( )  
 – Returns -  


---
- *getHeight*  
 public int **getHeight**( )  
 – Returns -  


---
- *getLocation*  
 public Point **getLocation**( )  
 – Returns -  


---
- *getParent*  
 public ActorContainer **getParent**( )  
 – Returns -  


---
- *getRootLocation*  
 public Point **getRootLocation**( )  
 – Returns -  


---
- *getShadow*  
 public int **getShadow**( )  
 – Returns -  


---
- *getSize*  
 public Dimension **getSize**( )  
 – Returns -  


---
- *getWidth*  
 public int **getWidth**( )  
 – Returns -  


---
- *getX*  
 public int **getX**( )  
 – Returns -  


---
- *getY*  
 public int **getY**( )  
 – Returns -  


---
- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – Usage  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – Parameters  
   \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – Usage  
   \* Paints the actors contained in the vector on this actor.  
 – Parameters  
   \* g -

- - \* actors -

---
- *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
    - Parameters
      - \* g -
      - \* backImage -
      - \* xx -
      - \* yy -
      - \* w -
      - \* h -

---
- *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )
    - Usage
      - \* Paints the shadow of the actor. Override this in the subclasses if needed.
    - Parameters
      - \* g -

---
- *setBackground*  
public void **setBackground**( java.awt.Color bgcolor )
    - Parameters
      - \* bgcolor -

---
- *setBorderWidth*  
public void **setBorderWidth**( int w )
    - Parameters
      - \* w -

---
- *setBounds*  
public void **setBounds**( int x, int y, int w, int h )
    - Parameters
      - \* x -
      - \* y -
      - \* w -
      - \* h -

---
- *setFont*  
public void **setFont**( java.awt.Font font )
    - Parameters
      - \* font -

---
- *setForeground*  
public void **setForeground**( java.awt.Color fgcolor )
    - Parameters
      - \* fgcolor -

---
- *setInsets*  
public void **setInsets**( java.awt.Insets insets )
    - Parameters
      - \* insets -

---
- *setLight*  
public void **setLight**( int light )



- **Parameters**
  - \* *light* -

---

- *setLocation*  
 public void setLocation( int x, int y )
  - **Parameters**
  - \* *x* -
  - \* *y* -

---

- *setLocation*  
 public void setLocation( java.awt.Point loc )
  - **Parameters**
  - \* *loc* -

---

- *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - **Parameters**
  - \* *parent* -

---

- *setShadow*  
 public void setShadow( int s )
  - **Parameters**
  - \* *s* -

---

- *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - **Parameters**
  - \* *si* -

---

- *setSize*  
 public void setSize( java.awt.Dimension d )
  - **Parameters**
  - \* *d* -

---

- *setSize*  
 public void setSize( int w, int h )
  - **Parameters**
  - \* *w* -
  - \* *h* -

### 16.2.35 CLASS Theater

---

This is the theatre component that is added in the left pane of the user interface and on which the program animation produced in the theater package is currently drawn.

#### DECLARATION

---

```
public class Theater
extends javax.swing.JComponent
implements ActorContainer
```

SERIALIZABLE FIELDS

---

- private Image backImage
  - Background image.
- private Image captScreen
  - Captured image of the screen, used on active mode for extra efficiency.
- private Graphics csg
  - Graphics object for captured image when the animation is going on.
- private boolean active
  - True, if the theatre is in active mode or captured. Active mode means that something is or is going to be animated. This means that the extra efficiency is needed and needless painting of all the actors is not done.
- private Vector pasAct
  - Vector of passive actors which are drawn in passive mode.
- private Vector actAct
  - Vector of active, moving actors which are drawn in active mode (during animation).
- private Actor highActor
  - Highlighted actor if any.
- private TheaterManager manager
  -
- private boolean showComponents
  - Variable is set true if there are other **JComponents** on the Theatre component. At the moment this happens only when input is requested. The state of the variable changes the operation of the **paint** method.

FIELDS

---

- private Image backImage
  - Background image.
- private Image captScreen
  - Captured image of the screen, used on active mode for extra efficiency.
- private Graphics csg
  - Graphics object for captured image when the animation is going on.
- private boolean active
  - True, if the theatre is in active mode or captured. Active mode means that something is or is going to be animated. This means that the extra efficiency is needed and needless painting of all the actors is not done.

- private Vector pasAct
  - Vector of passive actors which are drawn in passive mode.
- private Vector actAct
  - Vector of active, moving actors which are drawn in active mode (during animation).
- private Actor highActor
  - Highlighted actor if any.
- private TheaterManager manager
  -
- private boolean showComponents
  - Variable is set true if there are other JComponents on the Theatre component. At the moment this happens only when input is requested. The state of the variable changes the operation of the paint method.

## CONSTRUCTORS

---

- *Theater*  
 public Theater( )
  - **Usage**
    - \* Sets the opaque of the component to be true.
  - **See Also**
    - \* jeliot.theater.Theater.setOpaque(boolean)

## METHODS

---

- *addActor*  
 public void addActor( jeliot.theater.Actor actor )
  - **Parameters**
    - \* actor -

---
- *addPassive*  
 public void addPassive( jeliot.theater.Actor actor )
  - **Parameters**
    - \* actor -

---
- *capture*  
 public void capture( )
 

---
- *cleanUp*  
 public void cleanUp( )
 

---
- *flush*  
 public void flush( )
 

---

- *getActorAt*  
 public Actor **getActorAt**( int x, int y )  
 – **Parameters**  
   \* x -  
   \* y -  
 – **Returns** -  


---
- *getHeight*  
 public int **getHeight**( )  


---
- *getManager*  
 public TheaterManager **getManager**( )  
 – **Usage**  
   \* Returns the TheatreManager  
 – **Returns** - TheatreManager object  


---
- *getWidth*  
 public int **getWidth**( )  


---
- *isCaptured*  
 public boolean **isCaptured**( )  
 – **Returns** -  


---
- *paint*  
 public void **paint**( java.awt.Graphics g )  


---
- *paintActors*  
 private void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – **Usage**  
   \* Paints the actors contained in given vector.  
 – **Parameters**  
   \* g -  
   \* actors -  


---
- *paintBackground*  
 private void **paintBackground**( java.awt.Graphics g )  
 – **Usage**  
   \* Fills the background with background image.  
 – **Parameters**  
   \* g -  


---
- *paintCapturedScreen*  
 private void **paintCapturedScreen**( java.awt.Graphics g )  
 – **Usage**  
   \* Paints the image of captured screen.  
 – **Parameters**  
   \* g -

---

- *paintComponent*

```
public void paintComponent( java.awt.Graphics g )
```

- **Usage**

- \* Paints the theatre. If theatre is in active mode then a captured picture and only active actors are painted otherwise background, the passive, highlighted and active actors are painted.

- **Parameters**

- \* **g** - Everything is painted on the given Graphics object.

---

- *paintHighlight*

```
private void paintHighlight( java.awt.Graphics g )
```

- **Usage**

- \* Paints the highlight marker around highlighted actor.

- **Parameters**

- \* **g** -

---

- *passivate*

```
public void passivate( jeliot.theater.Actor actor )
```

- **Parameters**

- \* **actor** -

---

- *promote*

```
public void promote( jeliot.theater.Actor actor )
```

- **Parameters**

- \* **actor** -

---

- *release*

```
public void release( )
```

---

- *removeActor*

```
public void removeActor( jeliot.theater.Actor actor )
```

---

- *removePassive*

```
public void removePassive( jeliot.theater.Actor actor )
```

- **Parameters**

- \* **actor** -

---

- *setBackground*

```
public void setBackground( java.awt.Image backImage )
```

- **Usage**

- \* Sets the background image (**backImage**) of this theatre.

- **Parameters**

- \* **backImage** - Image for background.

---

- *setHighlightedActor*  
 public void **setHighlightedActor**( jeliot.theater.Actor actor )  
 – **Parameters**  
 \* actor -  


---
- *showComponents*  
 public void **showComponents**( boolean show )  
 – **Parameters**  
 \* show -  


---
- *updateCapture*  
 public void **updateCapture**( )

---

#### METHODS INHERITED FROM CLASS javax.swing.JComponent

---

- *\_paintImmediately*  
 void **\_paintImmediately**( int , int , int , int )  


---
- *<clinit>*  
 static void **<clinit>**( )  


---
- *addAncestorListener*  
 public void **addAncestorListener**( javax.swing.event.AncestorListener )  


---
- *addNotify*  
 public void **addNotify**( )  


---
- *addPropertyChangeListener*  
 public synchronized void **addPropertyChangeListener**(  
 java.beans.PropertyChangeListener )  


---
- *addPropertyChangeListener*  
 public synchronized void **addPropertyChangeListener**( java.lang.String ,  
 java.beans.PropertyChangeListener )  


---
- *addVetoableChangeListener*  
 public synchronized void **addVetoableChangeListener**(  
 java.beans.VetoableChangeListener )  


---
- *adjustPaintFlags*  
 private void **adjustPaintFlags**( )  


---
- *alwaysOnTop*  
 boolean **alwaysOnTop**( )  


---
- *checkIfChildObscuredBySibling*  
 boolean **checkIfChildObscuredBySibling**( )  


---
- *componentInputMapChanged*  
 void **componentInputMapChanged**( javax.swing.ComponentInputMap )  


---
- *computeVisibleRect*  
 static final void **computeVisibleRect**( java.awt.Component , java.awt.Rectangle )  


---
- *computeVisibleRect*  
 public void **computeVisibleRect**( java.awt.Rectangle )  


---
- *compWriteObjectNotify*  
 void **compWriteObjectNotify**( )  


---
- *contains*  
 public boolean **contains**( int , int )  


---

- *createToolTip*  
public JToolTip createToolTip( )
- *deregisterNextFocusableComponent*  
private void deregisterNextFocusableComponent( )
- *disable*  
public void disable( )
- *enable*  
public void enable( )
- *enableSerialization*  
void enableSerialization( )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , byte , byte )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , char , char )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , double , double )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , float , float )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , long , long )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *firePropertyChange*  
public void firePropertyChange( java.lang.String , short , short )
- *fireVetoableChange*  
protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getActionForKeyStroke*  
public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )
- *getActionMap*  
public final ActionMap getActionMap( )
- *getActionMap*  
final ActionMap getActionMap( boolean )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getAncestorListeners*  
public AncestorListener getAncestorListeners( )
- *getAutoscrolls*  
public boolean getAutoscrolls( )
- *getBorder*  
public Border getBorder( )

- *getBounds*  
public Rectangle getBounds( java.awt.Rectangle )
- *getClientProperties*  
private Dictionary getClientProperties( )
- *getClientProperty*  
public final Object getClientProperty( java.lang.Object )
- *getComponentGraphics*  
protected Graphics getComponentGraphics( java.awt.Graphics )
- *getConditionForKeyStroke*  
public int getConditionForKeyStroke( javax.swing.KeyStroke )
- *getCreatedDoubleBuffer*  
boolean getCreatedDoubleBuffer( boolean )
- *getDebugGraphicsOptions*  
public int getDebugGraphicsOptions( )
- *getDefaultLocale*  
public static Locale getDefaultLocale( )
- *getFlag*  
private boolean getFlag( int )
- *getGraphics*  
public Graphics getGraphics( )
- *getHeight*  
public int getHeight( )
- *getInputMap*  
public final InputMap getInputMap( )
- *getInputMap*  
public final InputMap getInputMap( int )
- *getInputMap*  
final InputMap getInputMap( int , boolean )
- *getInputVerifier*  
public InputVerifier getInputVerifier( )
- *getInsets*  
public Insets getInsets( )
- *getInsets*  
public Insets getInsets( java.awt.Insets )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getManagingFocusBackwardTraversalKeys*  
static Set getManagingFocusBackwardTraversalKeys( )
- *getManagingFocusForwardTraversalKeys*  
static Set getManagingFocusForwardTraversalKeys( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getNextFocusableComponent*  
public Component getNextFocusableComponent( )
- *getObscuredState*  
private int getObscuredState( int , int , int , int , int )



- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getRegisteredKeyStrokes*  
public KeyStroke **getRegisteredKeyStrokes**( )
- *getRootPane*  
public JRootPane **getRootPane**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getSuppressDropTarget*  
private static boolean **getSuppressDropTarget**( )
- *getToolTipLocation*  
public Point **getToolTipLocation**( java.awt.event.MouseEvent )
- *getToolTipText*  
public String **getToolTipText**( )
- *getToolTipText*  
public String **getToolTipText**( java.awt.event.MouseEvent )
- *getTopLevelAncestor*  
public Container **getTopLevelAncestor**( )
- *getTransferHandler*  
public TransferHandler **getTransferHandler**( )
- *getUIClassID*  
public String **getUIClassID**( )
- *getVerifyInputWhenFocusTarget*  
public boolean **getVerifyInputWhenFocusTarget**( )
- *getVetoableChangeListeners*  
public synchronized VetoableChangeListener **getVetoableChangeListeners**( )
- *getWidth*  
public int **getWidth**( )
- *getVisibleRect*  
public Rectangle **getVisibleRect**( )
- *getWriteObjCounter*  
static byte **getWriteObjCounter**( javax.swing.JComponent )
- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *grabFocus*  
public void **grabFocus**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isLightweightComponent*  
public static boolean **isLightweightComponent**( java.awt.Component )
- *isManagingFocus*  
public boolean **isManagingFocus**( )
- *isMaximumSizeSet*  
public boolean **isMaximumSizeSet**( )

- *isMinimumSizeSet*  
public boolean isMinimumSizeSet( )
- *isOpaque*  
public boolean isOpaque( )
- *isOptimizedDrawingEnabled*  
public boolean isOptimizedDrawingEnabled( )
- *isPaintingOrigin*  
boolean isPaintingOrigin( )
- *isPaintingTile*  
public boolean isPaintingTile( )
- *isPreferredSizeSet*  
public boolean isPreferredSizeSet( )
- *isRequestFocusEnabled*  
public boolean isRequestFocusEnabled( )
- *isValidateRoot*  
public boolean isValidateRoot( )
- *paint*  
public void paint( java.awt.Graphics )
- *paintBorder*  
protected void paintBorder( java.awt.Graphics )
- *paintChildren*  
protected void paintChildren( java.awt.Graphics )
- *paintComponent*  
protected void paintComponent( java.awt.Graphics )
- *paintDoubleBuffered*  
private boolean paintDoubleBuffered( javax.swing.JComponent , java.awt.Component , java.awt.Graphics , int , int , int , int )
- *paintImmediately*  
public void paintImmediately( int , int , int , int )
- *paintImmediately*  
public void paintImmediately( java.awt.Rectangle )
- *paintWithOffscreenBuffer*  
private void paintWithOffscreenBuffer( javax.swing.JComponent , java.awt.Graphics , int , int , int , int , int , java.awt.Image )
- *paramString*  
protected String paramString( )
- *print*  
public void print( java.awt.Graphics )
- *printAll*  
public void printAll( java.awt.Graphics )
- *printBorder*  
protected void printBorder( java.awt.Graphics )
- *printChildren*  
protected void printChildren( java.awt.Graphics )
- *printComponent*  
protected void printComponent( java.awt.Graphics )
- *processComponentKeyEvent*  
protected void processComponentKeyEvent( java.awt.event.KeyEvent )
- *processKeyBinding*  
protected boolean processKeyBinding( javax.swing.KeyStroke , java.awt.event.KeyEvent , int , boolean )

- *processKeyBindings*  
boolean processKeyBindings( java.awt.event.KeyEvent , boolean )
- *processKeyBindingsForAllComponents*  
static boolean processKeyBindingsForAllComponents( java.awt.event.KeyEvent , java.awt.Container , boolean )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent )
- *putClientProperty*  
public final void putClientProperty( java.lang.Object , java.lang.Object )
- *readObject*  
private void readObject( java.io.ObjectInputStream )
- *rectangleIsObscured*  
boolean rectangleIsObscured( int , int , int , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener , javax.swing.KeyStroke , int )
- *registerKeyboardAction*  
public void registerKeyboardAction( java.awt.event.ActionListener , java.lang.String , javax.swing.KeyStroke , int )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( )
- *registerNextFocusableComponent*  
private void registerNextFocusableComponent( java.awt.Component )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( boolean )
- *registerWithKeyboardManager*  
private void registerWithKeyboardManager( javax.swing.KeyStroke )
- *removeAncestorListener*  
public void removeAncestorListener( javax.swing.event.AncestorListener )
- *removeNotify*  
public void removeNotify( )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *removeVetoableChangeListener*  
public synchronized void removeVetoableChangeListener( java.beans.VetoableChangeListener )
- *repaint*  
public void repaint( long , int , int , int , int )
- *repaint*  
public void repaint( java.awt.Rectangle )
- *requestDefaultFocus*  
public boolean requestDefaultFocus( )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
public boolean requestFocus( boolean )

- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetKeyboardActions*  
public void resetKeyboardActions( )
- *reshape*  
public void reshape( int , int , int , int )
- *revalidate*  
public void revalidate( )
- *runInputVerifier*  
private boolean runInputVerifier( )
- *scrollRectToVisible*  
public void scrollRectToVisible( java.awt.Rectangle )
- *setActionMap*  
public final void setActionMap( javax.swing.ActionMap )
- *setAlignmentX*  
public void setAlignmentX( float )
- *setAlignmentY*  
public void setAlignmentY( float )
- *setAutoscrolls*  
public void setAutoscrolls( boolean )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBorder*  
public void setBorder( javax.swing.border.Border )
- *setCreatedDoubleBuffer*  
void setCreatedDoubleBuffer( boolean , boolean )
- *setDebugGraphicsOptions*  
public void setDebugGraphicsOptions( int )
- *setDefaultLocale*  
public static void setDefaultLocale( java.util.Locale )
- *setDoubleBuffered*  
public void setDoubleBuffered( boolean )
- *setEnabled*  
public void setEnabled( boolean )
- *setFlag*  
private void setFlag( int , boolean )
- *setFont*  
public void setFont( java.awt.Font )
- *setForeground*  
public void setForeground( java.awt.Color )
- *setInputMap*  
public final void setInputMap( int , javax.swing.InputMap )
- *setInputVerifier*  
public void setInputVerifier( javax.swing.InputVerifier )
- *setMaximumSize*  
public void setMaximumSize( java.awt.Dimension )
- *setMinimumSize*  
public void setMinimumSize( java.awt.Dimension )

- *setNextFocusableComponent*  
public void setNextFocusableComponent( java.awt.Component )
- *setOpaque*  
public void setOpaque( boolean )
- *setPaintingChild*  
void setPaintingChild( java.awt.Component )
- *setPreferredSize*  
public void setPreferredSize( java.awt.Dimension )
- *setRequestFocusEnabled*  
public void setRequestFocusEnabled( boolean )
- *setToolTipText*  
public void setToolTipText( java.lang.String )
- *setTransferHandler*  
public void setTransferHandler( javax.swing.TransferHandler )
- *setUI*  
protected void setUI( javax.swing.plaf.ComponentUI )
- *setVerifyInputWhenFocusTarget*  
public void setVerifyInputWhenFocusTarget( boolean )
- *setVisible*  
public void setVisible( boolean )
- *setWriteObjCounter*  
static void setWriteObjCounter( javax.swing.JComponent , byte )
- *shouldDebugGraphics*  
int shouldDebugGraphics( )
- *superProcessMouseEvent*  
void superProcessMouseEvent( java.awt.event.MouseEvent )
- *unregisterKeyboardAction*  
public void unregisterKeyboardAction( javax.swing.KeyStroke )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( )
- *unregisterWithKeyboardManager*  
private void unregisterWithKeyboardManager( javax.swing.KeyStroke )
- *update*  
public void update( java.awt.Graphics )
- *updateUI*  
public void updateUI( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

#### METHODS INHERITED FROM CLASS java.awt.Container

---

- *<clinit>*  
static void <clinit>( )
- *add*  
public Component add( java.awt.Component )
- *add*  
public Component add( java.awt.Component , int )
- *add*  
public void add( java.awt.Component , java.lang.Object )

- *add*  
public void add( java.awt.Component , java.lang.Object , int )
- *add*  
public Component add( java.lang.String , java.awt.Component )
- *addContainerListener*  
public synchronized void addContainerListener( java.awt.event.ContainerListener )
- *addImpl*  
protected void addImpl( java.awt.Component , java.lang.Object , int )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )
- *adjustDecendantsOnParent*  
void adjustDecendantsOnParent( int )
- *adjustDescendants*  
void adjustDescendants( int )
- *adjustListeningChildren*  
void adjustListeningChildren( long , int )
- *applyComponentOrientation*  
public void applyComponentOrientation( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean areFocusTraversalKeysSet( int )
- *checkGD*  
void checkGD( java.lang.String )
- *clearCurrentFocusCycleRootOnHide*  
void clearCurrentFocusCycleRootOnHide( )
- *clearMostRecentFocusOwnerOnHide*  
void clearMostRecentFocusOwnerOnHide( )
- *containsFocus*  
final boolean containsFocus( )
- *countComponents*  
public int countComponents( )
- *countHierarchyMembers*  
int countHierarchyMembers( )
- *createChildHierarchyEvents*  
void createChildHierarchyEvents( int , long , boolean )
- *createHierarchyEvents*  
int createHierarchyEvents( int , java.awt.Component , java.awt.Container , long , boolean )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchEventToSelf*  
void dispatchEventToSelf( java.awt.AWTEvent )
- *doLayout*  
public void doLayout( )

- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *findComponentAt*  
public Component findComponentAt( int , int )
- *findComponentAt*  
final Component findComponentAt( int , int , boolean )
- *findComponentAt*  
public Component findComponentAt( java.awt.Point )
- *findTraversalRoot*  
private Container findTraversalRoot( )
- *getAccessibleAt*  
Accessible getAccessibleAt( java.awt.Point )
- *getAccessibleChild*  
Accessible getAccessibleChild( int )
- *getAccessibleChildrenCount*  
int getAccessibleChildrenCount( )
- *getAlignmentX*  
public float getAlignmentX( )
- *getAlignmentY*  
public float getAlignmentY( )
- *getComponent*  
public Component getComponent( int )
- *getComponentAt*  
public Component getComponentAt( int , int )
- *getComponentAt*  
public Component getComponentAt( java.awt.Point )
- *getComponentCount*  
public int getComponentCount( )
- *getComponents\_NoClientCode*  
final Component getComponents\_NoClientCode( )
- *getComponents*  
public Component getComponents( )
- *getContainerListeners*  
public synchronized ContainerListener getContainerListeners( )
- *getDropTargetEventTarget*  
Component getDropTargetEventTarget( int , int , boolean )
- *getFocusTraversalKeys*  
public Set getFocusTraversalKeys( int )
- *getFocusTraversalPolicy*  
public FocusTraversalPolicy getFocusTraversalPolicy( )
- *getInsets*  
public Insets getInsets( )
- *getLayout*  
public LayoutManager getLayout( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )

- *getMouseEventTarget*  
Component **getMouseEventTarget**( int , int , boolean )
- *getMouseEventTarget*  
private Component **getMouseEventTarget**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean )
- *getMouseEventTargetImpl*  
private Component **getMouseEventTargetImpl**( int , int , boolean ,  
java.awt.Container.EventTargetFilter , boolean , boolean )
- *getPreferredSize*  
public Dimension **getPreferredSize**( )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *insets*  
public Insets **insets**( )
- *invalidate*  
public void **invalidate**( )
- *invalidateTree*  
void **invalidateTree**( )
- *isAncestorOf*  
public boolean **isAncestorOf**( java.awt.Component )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( )
- *isFocusCycleRoot*  
public boolean **isFocusCycleRoot**( java.awt.Container )
- *isFocusTraversalPolicySet*  
public boolean **isFocusTraversalPolicySet**( )
- *isParentOf*  
boolean **isParentOf**( java.awt.Component )
- *layout*  
public void **layout**( )
- *lightweightPaint*  
void **lightweightPaint**( java.awt.Graphics )
- *lightweightPrint*  
void **lightweightPrint**( java.awt.Graphics )
- *list*  
public void **list**( java.io.PrintStream , int )
- *list*  
public void **list**( java.io.PrintWriter , int )
- *locate*  
public Component **locate**( int , int )
- *minimumSize*  
public Dimension **minimumSize**( )
- *nextFocusHelper*  
boolean **nextFocusHelper**( )
- *numListening*  
int **numListening**( long )
- *paint*  
public void **paint**( java.awt.Graphics )



- *paintComponents*  
public void **paintComponents**( java.awt.Graphics    )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics    )
- *paramString*  
protected String **paramString**(    )
- *postProcessKeyEvent*  
void **postProcessKeyEvent**( java.awt.event.KeyEvent    )
- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**(    )
- *preferredSize*  
public Dimension **preferredSize**(    )
- *preProcessKeyEvent*  
void **preProcessKeyEvent**( java.awt.event.KeyEvent    )
- *print*  
public void **print**( java.awt.Graphics    )
- *printComponents*  
public void **printComponents**( java.awt.Graphics    )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics    )
- *processContainerEvent*  
protected void **processContainerEvent**( java.awt.event.ContainerEvent    )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent    )
- *proxyEnableEvents*  
void **proxyEnableEvents**( long    )
- *readObject*  
private void **readObject**( java.io.ObjectInputStream    )
- *remove*  
public void **remove**( java.awt.Component    )
- *remove*  
public void **remove**( int    )
- *removeAll*  
public void **removeAll**(    )
- *removeContainerListener*  
public synchronized void **removeContainerListener**( java.awt.event.ContainerListener  
  )
- *removeNotify*  
public void **removeNotify**(    )
- *setFocusCycleRoot*  
public void **setFocusCycleRoot**( boolean    )
- *setFocusTraversalKeys*  
public void **setFocusTraversalKeys**( int    , java.util.Set    )
- *setFocusTraversalPolicy*  
public void **setFocusTraversalPolicy**( java.awt.FocusTraversalPolicy    )
- *setFont*  
public void **setFont**( java.awt.Font    )
- *setLayout*  
public void **setLayout**( java.awt.LayoutManager    )
- *setZOrder*  
void **setZOrder**( java.awt.Component    , int    )

- *transferFocusBackward*  
public void transferFocusBackward( )
- *transferFocusDownCycle*  
public void transferFocusDownCycle( )
- *update*  
public void update( java.awt.Graphics )
- *validate*  
public void validate( )
- *validateTree*  
protected void validateTree( )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream )

---

#### METHODS INHERITED FROM CLASS java.awt.Component

---

- *<clinit>*  
static void <clinit>( )
- *action*  
public boolean action( java.awt.Event , java.lang.Object )
- *add*  
public synchronized void add( java.awt.PopupMenu )
- *addComponentListener*  
public synchronized void addComponentListener( java.awt.event.ComponentListener )
- *addFocusListener*  
public synchronized void addFocusListener( java.awt.event.FocusListener )
- *addHierarchyBoundsListener*  
public void addHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener )
- *addHierarchyListener*  
public void addHierarchyListener( java.awt.event.HierarchyListener )
- *addInputMethodListener*  
public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )
- *addKeyListener*  
public synchronized void addKeyListener( java.awt.event.KeyListener )
- *addMouseListener*  
public synchronized void addMouseListener( java.awt.event.MouseListener )
- *addMouseMotionListener*  
public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )
- *addMouseWheelListener*  
public synchronized void addMouseWheelListener( java.awt.event.MouseWheelListener )
- *addNotify*  
public void addNotify( )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )
- *addPropertyChangeListener*  
public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )

- *adjustListeningChildrenOnParent*  
void **adjustListeningChildrenOnParent**( long , int )
- *applyComponentOrientation*  
public void **applyComponentOrientation**( java.awt.ComponentOrientation )
- *areFocusTraversalKeysSet*  
public boolean **areFocusTraversalKeysSet**( int )
- *areInputMethodsEnabled*  
boolean **areInputMethodsEnabled**( )
- *autoProcessMouseWheel*  
void **autoProcessMouseWheel**( java.awt.event.MouseWheelEvent )
- *autoTransferFocus*  
final void **autoTransferFocus**( boolean )
- *bounds*  
public Rectangle **bounds**( )
- *checkGD*  
void **checkGD**( java.lang.String )
- *checkImage*  
public int **checkImage**( java.awt.Image , java.awt.image.ImageObserver )
- *checkImage*  
public int **checkImage**( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *checkWindowClosingException*  
boolean **checkWindowClosingException**( )
- *clearCurrentFocusCycleRootOnHide*  
void **clearCurrentFocusCycleRootOnHide**( )
- *clearMostRecentFocusOwnerOnHide*  
void **clearMostRecentFocusOwnerOnHide**( )
- *coalesceEvents*  
protected AWTEvent **coalesceEvents**( java.awt.AWTEvent , java.awt.AWTEvent )
- *constructComponentName*  
String **constructComponentName**( )
- *contains*  
public boolean **contains**( int , int )
- *contains*  
public boolean **contains**( java.awt.Point )
- *containsFocus*  
boolean **containsFocus**( )
- *countHierarchyMembers*  
int **countHierarchyMembers**( )
- *createBufferStrategy*  
void **createBufferStrategy**( int )
- *createBufferStrategy*  
void **createBufferStrategy**( int , java.awt.BufferCapabilities )
- *createChildHierarchyEvents*  
void **createChildHierarchyEvents**( int , long , boolean )
- *createHierarchyEvents*  
int **createHierarchyEvents**( int , java.awt.Component , java.awt.Container , long , boolean )
- *createImage*  
public Image **createImage**( java.awt.image.ImageProducer )

- *createImage*  
public Image createImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int )
- *createVolatileImage*  
public VolatileImage createVolatileImage( int , int , java.awt.ImageCapabilities )
- *deliverEvent*  
public void deliverEvent( java.awt.Event )
- *disable*  
public void disable( )
- *disableEvents*  
protected final void disableEvents( long )
- *dispatchEvent*  
public final void dispatchEvent( java.awt.AWTEvent )
- *dispatchEventImpl*  
void dispatchEventImpl( java.awt.AWTEvent )
- *dispatchMouseWheelToAncestor*  
boolean dispatchMouseWheelToAncestor( java.awt.event.MouseWheelEvent )
- *doAutoTransfer*  
private void doAutoTransfer( boolean )
- *doLayout*  
public void doLayout( )
- *enable*  
public void enable( )
- *enable*  
public void enable( boolean )
- *enableEvents*  
protected final void enableEvents( long )
- *enableInputMethods*  
public void enableInputMethods( boolean )
- *eventEnabled*  
boolean eventEnabled( java.awt.AWTEvent )
- *eventTypeEnabled*  
boolean eventTypeEnabled( int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , boolean , boolean )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , int , int )
- *firePropertyChange*  
protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )
- *getAccessibleContext*  
public AccessibleContext getAccessibleContext( )
- *getAccessibleIndexInParent*  
int getAccessibleIndexInParent( )
- *getAccessibleStateSet*  
AccessibleStateSet getAccessibleStateSet( )
- *getAlignmentX*  
public float getAlignmentX( )

- *getAlignmentY*  
public float **getAlignmentY**( )
- *getBackBuffer*  
Image **getBackBuffer**( )
- *getBackground*  
public Color **getBackground**( )
- *getBounds*  
public Rectangle **getBounds**( )
- *getBounds*  
public Rectangle **getBounds**( java.awt.Rectangle )
- *getBufferStrategy*  
BufferStrategy **getBufferStrategy**( )
- *getColorModel*  
public ColorModel **getColorModel**( )
- *getComponentAt*  
public Component **getComponentAt**( int , int )
- *getComponentAt*  
public Component **getComponentAt**( java.awt.Point )
- *getComponentListeners*  
public synchronized ComponentListener **getComponentListeners**( )
- *getComponentOrientation*  
public ComponentOrientation **getComponentOrientation**( )
- *getCursor*  
public Cursor **getCursor**( )
- *getDropTarget*  
public synchronized DropTarget **getDropTarget**( )
- *getFocusCycleRootAncestor*  
public Container **getFocusCycleRootAncestor**( )
- *getFocusListeners*  
public synchronized FocusListener **getFocusListeners**( )
- *getFocusTraversalKeys\_NoIDCheck*  
final Set **getFocusTraversalKeys\_NoIDCheck**( int )
- *getFocusTraversalKeys*  
public Set **getFocusTraversalKeys**( int )
- *getFocusTraversalKeysEnabled*  
public boolean **getFocusTraversalKeysEnabled**( )
- *getFont\_NoClientCode*  
final Font **getFont\_NoClientCode**( )
- *getFont*  
public Font **getFont**( )
- *getFontMetrics*  
public FontMetrics **getFontMetrics**( java.awt.Font )
- *getForeground*  
public Color **getForeground**( )
- *getGraphics*  
public Graphics **getGraphics**( )
- *getGraphicsConfiguration*  
public GraphicsConfiguration **getGraphicsConfiguration**( )
- *getHeight*  
public int **getHeight**( )

- *getHierarchyBoundsListeners*  
public synchronized HierarchyBoundsListener getHierarchyBoundsListeners( )
- *getHierarchyListeners*  
public synchronized HierarchyListener getHierarchyListeners( )
- *getIgnoreRepaint*  
public boolean getIgnoreRepaint( )
- *getInputContext*  
public InputContext getInputContext( )
- *getInputMethodListeners*  
public synchronized InputMethodListener getInputMethodListeners( )
- *getInputMethodRequests*  
public InputMethodRequests getInputMethodRequests( )
- *getKeyListeners*  
public synchronized KeyListener getKeyListeners( )
- *getListeners*  
public EventListener getListeners( java.lang.Class )
- *getLocale*  
public Locale getLocale( )
- *getLocation*  
public Point getLocation( )
- *getLocation*  
public Point getLocation( java.awt.Point )
- *getLocationOnScreen\_NoTreeLock*  
final Point getLocationOnScreen\_NoTreeLock( )
- *getLocationOnScreen*  
public Point getLocationOnScreen( )
- *getMaximumSize*  
public Dimension getMaximumSize( )
- *getMinimumSize*  
public Dimension getMinimumSize( )
- *getMouseListeners*  
public synchronized MouseListener getMouseListeners( )
- *getMouseMotionListeners*  
public synchronized MouseMotionListener getMouseMotionListeners( )
- *getMouseWheelListeners*  
public synchronized MouseWheelListener getMouseWheelListeners( )
- *getName*  
public String getName( )
- *getNativeContainer*  
Container getNativeContainer( )
- *getParent\_NoClientCode*  
final Container getParent\_NoClientCode( )
- *getParent*  
public Container getParent( )
- *getPeer*  
public ComponentPeer getPeer( )
- *getPreferredSize*  
public Dimension getPreferredSize( )
- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener getPropertyChangeListeners( )

- *getPropertyChangeListeners*  
public synchronized PropertyChangeListener **getPropertyChangeListeners**(  
java.lang.String )
- *getSize*  
public Dimension **getSize**( )
- *getSize*  
public Dimension **getSize**( java.awt.Dimension )
- *getToolkit*  
public Toolkit **getToolkit**( )
- *getToolkitImpl*  
final Toolkit **getToolkitImpl**( )
- *getTreeLock*  
public final Object **getTreeLock**( )
- *getWidth*  
public int **getWidth**( )
- *getX*  
public int **getX**( )
- *getY*  
public int **getY**( )
- *gotFocus*  
public boolean **gotFocus**( java.awt.Event , java.lang.Object )
- *handleEvent*  
public boolean **handleEvent**( java.awt.Event )
- *hasFocus*  
public boolean **hasFocus**( )
- *hide*  
public void **hide**( )
- *imageUpdate*  
public boolean **imageUpdate**( java.awt.Image , int , int , int , int , int )
- *initializeFocusTraversalKeys*  
void **initializeFocusTraversalKeys**( )
- *initIDs*  
private static native void **initIDs**( )
- *inside*  
public boolean **inside**( int , int )
- *invalidate*  
public void **invalidate**( )
- *isBackgroundSet*  
public boolean **isBackgroundSet**( )
- *isCursorSet*  
public boolean **isCursorSet**( )
- *isDisplayable*  
public boolean **isDisplayable**( )
- *isDoubleBuffered*  
public boolean **isDoubleBuffered**( )
- *isEnabled*  
public boolean **isEnabled**( )
- *isEnabledImpl*  
final boolean **isEnabledImpl**( )
- *isFocusable*  
public boolean **isFocusable**( )

- *isFocusCycleRoot*  
public boolean isFocusCycleRoot( java.awt.Container )
- *isFocusOwner*  
public boolean isFocusOwner( )
- *isFocusTraversable*  
public boolean isFocusTraversable( )
- *isFocusTraversableOverridden*  
final boolean isFocusTraversableOverridden( )
- *isFontSet*  
public boolean isFontSet( )
- *isForegroundSet*  
public boolean isForegroundSet( )
- *isLightweight*  
public boolean isLightweight( )
- *isOpaque*  
public boolean isOpaque( )
- *isRecursivelyVisible*  
boolean isRecursivelyVisible( )
- *isShowing*  
public boolean isShowing( )
- *isValid*  
public boolean isValid( )
- *isVisible*  
public boolean isVisible( )
- *keyDown*  
public boolean keyDown( java.awt.Event , int )
- *keyUp*  
public boolean keyUp( java.awt.Event , int )
- *layout*  
public void layout( )
- *lightweightPaint*  
void lightweightPaint( java.awt.Graphics )
- *lightweightPrint*  
void lightweightPrint( java.awt.Graphics )
- *list*  
public void list( )
- *list*  
public void list( java.io.PrintStream )
- *list*  
public void list( java.io.PrintStream , int )
- *list*  
public void list( java.io.PrintWriter )
- *list*  
public void list( java.io.PrintWriter , int )
- *locate*  
public Component locate( int , int )
- *location*  
public Point location( )
- *lostFocus*  
public boolean lostFocus( java.awt.Event , java.lang.Object )



- *minimumSize*  
public Dimension **minimumSize**( )
- *mouseDown*  
public boolean **mouseDown**( java.awt.Event , int , int )
- *mouseDrag*  
public boolean **mouseDrag**( java.awt.Event , int , int )
- *mouseEnter*  
public boolean **mouseEnter**( java.awt.Event , int , int )
- *mouseExit*  
public boolean **mouseExit**( java.awt.Event , int , int )
- *mouseMove*  
public boolean **mouseMove**( java.awt.Event , int , int )
- *mouseUp*  
public boolean **mouseUp**( java.awt.Event , int , int )
- *move*  
public void **move**( int , int )
- *nextFocus*  
public void **nextFocus**( )
- *nextFocusHelper*  
boolean **nextFocusHelper**( )
- *numListening*  
int **numListening**( long )
- *paint*  
public void **paint**( java.awt.Graphics )
- *paintAll*  
public void **paintAll**( java.awt.Graphics )
- *paintHeavyweightComponents*  
void **paintHeavyweightComponents**( java.awt.Graphics )
- *paramString*  
protected String **paramString**( )
- *postEvent*  
public boolean **postEvent**( java.awt.Event )
- *postsOldMouseEvents*  
boolean **postsOldMouseEvents**( )
- *preferredSize*  
public Dimension **preferredSize**( )
- *prepareImage*  
public boolean **prepareImage**( java.awt.Image , java.awt.image.ImageObserver )
- *prepareImage*  
public boolean **prepareImage**( java.awt.Image , int , int , java.awt.image.ImageObserver )
- *print*  
public void **print**( java.awt.Graphics )
- *printAll*  
public void **printAll**( java.awt.Graphics )
- *printHeavyweightComponents*  
void **printHeavyweightComponents**( java.awt.Graphics )
- *processComponentEvent*  
protected void **processComponentEvent**( java.awt.event.ComponentEvent )
- *processEvent*  
protected void **processEvent**( java.awt.AWTEvent )

- *processFocusEvent*  
protected void processFocusEvent( java.awt.event.FocusEvent    )
- *processHierarchyBoundsEvent*  
protected void processHierarchyBoundsEvent( java.awt.event.HierarchyEvent    )
- *processHierarchyEvent*  
protected void processHierarchyEvent( java.awt.event.HierarchyEvent    )
- *processInputMethodEvent*  
protected void processInputMethodEvent( java.awt.event.InputMethodEvent    )
- *processKeyEvent*  
protected void processKeyEvent( java.awt.event.KeyEvent    )
- *processMouseEvent*  
protected void processMouseEvent( java.awt.event.MouseEvent    )
- *processMouseMotionEvent*  
protected void processMouseMotionEvent( java.awt.event.MouseEvent    )
- *processMouseWheelEvent*  
protected void processMouseWheelEvent( java.awt.event.MouseWheelEvent    )
- *readObject*  
private void readObject( java.io.ObjectInputStream    )
- *remove*  
public synchronized void remove( java.awt.MenuComponent    )
- *removeComponentListener*  
public synchronized void removeComponentListener( java.awt.event.ComponentListener    )
- *removeFocusListener*  
public synchronized void removeFocusListener( java.awt.event.FocusListener    )
- *removeHierarchyBoundsListener*  
public void removeHierarchyBoundsListener( java.awt.event.HierarchyBoundsListener    )
- *removeHierarchyListener*  
public void removeHierarchyListener( java.awt.event.HierarchyListener    )
- *removeInputMethodListener*  
public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener    )
- *removeKeyListener*  
public synchronized void removeKeyListener( java.awt.event.KeyListener    )
- *removeMouseListener*  
public synchronized void removeMouseListener( java.awt.event.MouseListener    )
- *removeMouseMotionListener*  
public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener    )
- *removeMouseWheelListener*  
public synchronized void removeMouseWheelListener( java.awt.event.MouseWheelListener    )
- *removeNotify*  
public void removeNotify(    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener    )
- *removePropertyChangeListener*  
public synchronized void removePropertyChangeListener( java.lang.String    , java.beans.PropertyChangeListener    )

- *repaint*  
public void repaint( )
- *repaint*  
public void repaint( int , int , int , int )
- *repaint*  
public void repaint( long )
- *repaint*  
public void repaint( long , int , int , int , int )
- *requestFocus*  
public void requestFocus( )
- *requestFocus*  
protected boolean requestFocus( boolean )
- *requestFocusHelper*  
final boolean requestFocusHelper( boolean , boolean )
- *requestFocusInWindow*  
public boolean requestFocusInWindow( )
- *requestFocusInWindow*  
protected boolean requestFocusInWindow( boolean )
- *resetGC*  
void resetGC( )
- *reshape*  
public void reshape( int , int , int , int )
- *resize*  
public void resize( java.awt.Dimension )
- *resize*  
public void resize( int , int )
- *setBackground*  
public void setBackground( java.awt.Color )
- *setBounds*  
public void setBounds( int , int , int , int )
- *setBounds*  
public void setBounds( java.awt.Rectangle )
- *setComponentOrientation*  
public void setComponentOrientation( java.awt.ComponentOrientation )
- *setCursor*  
public void setCursor( java.awt.Cursor )
- *setDropTarget*  
public synchronized void setDropTarget( java.awt.dnd.DropTarget )
- *setEnabled*  
public void setEnabled( boolean )
- *setFocusable*  
public void setFocusable( boolean )
- *setFocusTraversalKeys\_NoIDCheck*  
final void setFocusTraversalKeys\_NoIDCheck( int , java.util.Set )
- *setFocusTraversalKeys*  
public void setFocusTraversalKeys( int , java.util.Set )
- *setFocusTraversalKeysEnabled*  
public void setFocusTraversalKeysEnabled( boolean )
- *setFont*  
public void setFont( java.awt.Font )

- *setForeground*  
public void setForeground( java.awt.Color    )
- *setGCFromPeer*  
void setGCFromPeer(    )
- *setIgnoreRepaint*  
public void setIgnoreRepaint( boolean    )
- *setLocale*  
public void setLocale( java.util.Locale    )
- *setLocation*  
public void setLocation( int    , int    )
- *setLocation*  
public void setLocation( java.awt.Point    )
- *setName*  
public void setName( java.lang.String    )
- *setSize*  
public void setSize( java.awt.Dimension    )
- *setSize*  
public void setSize( int    , int    )
- *setVisible*  
public void setVisible( boolean    )
- *show*  
public void show(    )
- *show*  
public void show( boolean    )
- *size*  
public Dimension size(    )
- *toString*  
public String toString(    )
- *transferFocus*  
public void transferFocus(    )
- *transferFocusBackward*  
public void transferFocusBackward(    )
- *transferFocusUpCycle*  
public void transferFocusUpCycle(    )
- *update*  
public void update( java.awt.Graphics    )
- *updateCursorImmediately*  
final void updateCursorImmediately(    )
- *validate*  
public void validate(    )
- *writeObject*  
private void writeObject( java.io.ObjectOutputStream    )

### 16.2.36 CLASS TheaterManager

---

TheaterManager allocates the space for all InstanceActors, MethodStages, Scratches and constants (ConstantBox), and also listens the Theater component for resizes so the the allocation of the space is valid after resizing of the Theater component.

## DECLARATION

---

```
public class TheaterManager
extends java.lang.Object
implements java.awt.event.ComponentListener
```

## FIELDS

- 
- private Point methodStagePoints
    - Contains Points that are set as the rootlocations of the new MethodStage instances. They are circulated so that after the last value of the table the first value is used again.
  - private Theater theatre
    - Reference to the current Theatre instance. Reference is need for two reasons: Firstly, to set this TheatreManager instance as the ComponentListener of the Theatre and then assign the actors in correct places when the Theatre (**JComponent**) resized. Secondly, for inserting new actors to the passive (**pasAct**) and active (**actAct**) actors.
  - private Stack methods
    -
  - private Vector objects
    -
  - private Vector scratches
    -
  - private ConstantBox constantBox
    -
  - private int maxMethodStageX
    -
  - private Hashtable reservations
    -
  - private Point lrCorner
    -
  - private int minInstanceY
    -
  - private int minInstanceX
    -

CONSTRUCTORS

---

- *TheaterManager*  
 public **TheaterManager**( jeliot.theater.Theater theatre )  
 – **Parameters**  
 \* theatre -

METHODS

---

- *addScratch*  
 public void **addScratch**( jeliot.theater.Scratch scratch )  
 – **Parameters**  
 \* scratch -
- *bind*  
 public void **bind**( jeliot.theater.InstanceActor actor )  
 – **Parameters**  
 \* actor -
- *bind*  
 public void **bind**( jeliot.theater.MethodStage stage )  
 – **Parameters**  
 \* stage -
- *cleanUp*  
 public void **cleanUp**( )
- *componentHidden*  
 public void **componentHidden**( java.awt.event.ComponentEvent e )
- *componentMoved*  
 public void **componentMoved**( java.awt.event.ComponentEvent e )
- *componentResized*  
 public void **componentResized**( java.awt.event.ComponentEvent e )  
 – **Usage**  
 \* Called, when the theatre object is resized. Rearranges the theatre after resizing.  
 – **See Also**  
 \*  
 java.awt.event.ComponentListener.componentResized(java.awt.event.ComponentEvent e)
- *componentShown*  
 public void **componentShown**( java.awt.event.ComponentEvent e )
- *getConstantPositionY*  
 public int **getConstantPositionY**( )

– **Returns -**

- 
- *getMinInstanceX*  
public int **getMinInstanceX**( )

– **Returns -**

- 
- *getMinInstanceY*  
public int **getMinInstanceY**( )

– **Returns -**

- 
- *getOutputPoint*  
public Point **getOutputPoint**( )

– **Returns -**

- 
- *getScratchPositionX*  
public int **getScratchPositionX**( )

– **Returns -**

- 
- *positionConstantBox*  
private void **positionConstantBox**( )

- 
- *positionObjects*  
public void **positionObjects**( java.awt.Point **from**, java.awt.Point **to** )

– **Parameters**

- \* **from** -
- \* **to** -

- 
- *removeInstance*  
public void **removeInstance**( jeliot.theater.InstanceActor **actor** )

– **Parameters**

- \* **actor** -

- 
- *removeMethodStage*  
public void **removeMethodStage**( jeliot.theater.MethodStage **stage** )

– **Parameters**

- \* **stage** -

- 
- *removeScratch*  
public void **removeScratch**( jeliot.theater.Scratch **scratch** )

– **Parameters**

- \* **scratch** -

- 
- *reserve*  
public Point **reserve**( jeliot.theater.InstanceActor **actor** )

– **Parameters**

- \* **actor** -

---

– **Returns** -

---

- *reserve*

```
public Point reserve( jeliot.theater.MethodStage stage )
```

– **Parameters**

\* stage -

– **Returns** -

---

- *setConstantBox*

```
public void setConstantBox( jeliot.theater.ConstantBox cbox )
```

– **Parameters**

\* cbox -

---

- *setLinesAndText*

```
public void setLinesAndText( jeliot.theater.LinesAndText lat )
```

– **Usage**

\* Draws the lines separating different areas and writes texts on them.

– **Parameters**

\* lat -

### 16.2.37 CLASS ThreadController

---

**ThreadController** allows the execution of the **Runnable** object controlled by it to be paused and resumed in a safe way. The controller gets a **Runnable** object in its constructor. After it has been constructed, the controller can be called to start or pause the execution of its runnable.

Calling the **pause** method does not pause the execution immediately, but only when the **checkPoint** method is next called in the controlled thread.

**Warning!** **ThreadController** does not check that the **checkPoint** method is called from correct thread.

#### DECLARATION

---

```
public class ThreadController
extends java.lang.Object
```

#### FIELDS

---

- `private static final int RUNNING`  
– Possible states of the controller.
- `private static final int PAUSEREQ`



- Possible states of the controller.
- `private static final int PAUSED`
  - Possible states of the controller.
- `private int status`
  - Current state of the controller.
- `private Runnable runner`
  - The Runnable object controled by this controller.
- `private Thread thread`
  - A thread in which the Runnable is executed.

## CONSTRUCTORS

---

- *ThreadController*  
`public ThreadController( java.lang.Runnable runner )`
  - **Usage**
    - \* Constructs a new controller for given Runnable.
  - **Parameters**
    - \* `runner` -

## METHODS

---

- *checkPoint*  
`public synchronized void checkPoint( )`
  - **Usage**
    - \* Calling the checkpoint(Controlled) method with null actual parameter value.
- *checkPoint*  
`public synchronized void checkPoint( jeliot.theater.Controlled cont )`
  - **Usage**
    - \* Pauses the execution, if pause() method has been called since previous checkpoint.
  - **Parameters**
    - \* `cont` -
- *pause*  
`public synchronized void pause( )`
  - **Usage**
    - \* Instructs the controller to pause execution in next check point.
- *start*  
`public synchronized void start( )`
  - **Usage**
    - \* Starts or resumes the Runnable immediately in its own thread.

### 16.2.38 CLASS Trace

---

The trace that an actor leaves when moving. Currently this is just implemented but not used.

## DECLARATION

---

```
public class Trace
extends java.lang.Object
```

## CONSTRUCTORS

- *Trace*  
public **Trace**( )

## METHODS

- *paint*  
public void **paint**( java.awt.Graphics g )  
– **Parameters**  
\* g -
- *putTrace*  
public void **putTrace**( int x, int y )  
– **Parameters**  
\* x -  
\* y -

## 16.2.39 CLASS ValueActor

---

**ValueActor** is an actor that represents graphically the language construct **Value**. The **Value**'s type is represented by the colors of the **ValueActor** and the **Value**'s value is printed out as the String representation of the value.

## DECLARATION

---

```
public class ValueActor
extends jeliot.theater.Actor
```

## FIELDS

- 
- private static ResourceBundle bundle  
– The resource bundle for theater package
  - private int namex  
–
  - private int namey  
–
  - private int swidth  
–
  - private int margin  
–

CONSTRUCTORS

---

- *ValueActor*  
`public ValueActor( )`

METHODS

---

- *calcLabelPosition*  
`protected void calcLabelPosition( )`
- *calculateSize*  
`public void calculateSize( )`
- *getLabel*  
`public String getLabel( )`  
  
– Returns -
- *getPreferredSize*  
`public Dimension getPreferredSize( )`  
  
– Returns -
- *paintActor*  
`public void paintActor( java.awt.Graphics g )`
- *paintValue*  
`public void paintValue( java.awt.Graphics g )`  
  
– Parameters  
\* g -
- *setBounds*  
`public void setBounds( int x, int y, int w, int h )`
- *setLabel*  
`public void setLabel( java.lang.String valstr )`  
  
– Parameters  
\* valstr -

METHODS INHERITED FROM CLASS jeliot.theater.Actor

---

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`  
  
– Usage  
\* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
- Parameters  
\* loc -
- Returns -
- *calculateSize*  
`public void calculateSize( )`
- *clone*  
`public Object clone( )`

- *fly*  
 public Animation fly( java.awt.Point p )  
 – Parameters  
   \* p -  
 – Returns -  


---
- *fly*  
 public Animation fly( java.awt.Point p, int shadow )  
 – Usage  
   \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.  
 – Parameters  
   \* p -  
   \* shadow -  
 – Returns -  


---
- *getActorAt*  
 public Actor getActorAt( int x, int y )  
 – Parameters  
   \* x -  
   \* y -  
 – Returns -  


---
- *getBackground*  
 public Color getBackground( )  
 – Returns -  


---
- *getFont*  
 public Font getFont( )  
 – Returns -  


---
- *getFontMetrics*  
 protected FontMetrics getFontMetrics( )  
 – Returns -  


---
- *getForeground*  
 public Color getForeground( )  
 – Returns -  


---
- *getHeight*  
 public int getHeight( )  
 – Returns -  


---
- *getLocation*  
 public Point getLocation( )  
 – Returns -  


---
- *getParent*  
 public ActorContainer getParent( )  
 – Returns -  


---
- *getRootLocation*  
 public Point getRootLocation( )

- **Returns -**

---

  - *getShadow*  
public int **getShadow**( )
  - **Returns -**

---

  - *getSize*  
public Dimension **getSize**( )
  - **Returns -**

---

  - *getWidth*  
public int **getWidth**( )
  - **Returns -**

---

  - *getX*  
public int **getX**( )
  - **Returns -**

---

  - *getY*  
public int **getY**( )
  - **Returns -**

---

- *paintActor*  
public void **paintActor**( java.awt.Graphics g )
  - **Usage**
    - \* Paints the actor on the given Graphics instance. Override this in subclasses.
  - **Parameters**
    - \* g - The Graphics object

---

- *paintActors*  
protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )
  - **Usage**
    - \* Paints the actors contained in the vector on this actor.
  - **Parameters**
    - \* g -
    - \* actors -

---

- *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
  - **Parameters**
    - \* g -
    - \* backImage -
    - \* xx -
    - \* yy -
    - \* w -
    - \* h -

---

- *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )
  - **Usage**
    - \* Paints the shadow of the actor. Override this in the subclasses if needed.
  - **Parameters**
    - \* g -

- 
- *setBackground*  
 public void setBackground( java.awt.Color bgcolor )  
 – Parameters  
   \* bgcolor -

---

  - *setBorderWidth*  
 public void setBorderWidth( int w )  
 – Parameters  
   \* w -

---

  - *setBounds*  
 public void setBounds( int x, int y, int w, int h )  
 – Parameters  
   \* x -  
   \* y -  
   \* w -  
   \* h -

---

  - *setFont*  
 public void setFont( java.awt.Font font )  
 – Parameters  
   \* font -

---

  - *setForeground*  
 public void setForeground( java.awt.Color fgcolor )  
 – Parameters  
   \* fgcolor -

---

  - *setInsets*  
 public void setInsets( java.awt.Insets insets )  
 – Parameters  
   \* insets -

---

  - *setLight*  
 public void setLight( int light )  
 – Parameters  
   \* light -

---

  - *setLocation*  
 public void setLocation( int x, int y )  
 – Parameters  
   \* x -  
   \* y -

---

  - *setLocation*  
 public void setLocation( java.awt.Point loc )  
 – Parameters  
   \* loc -

---

  - *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )  
 – Parameters  
   \* parent -

- 
- *setShadow*  
 public void **setShadow**( int s )  
 – Parameters  
 \* s -

---

  - *setShadowImage*  
 public static void **setShadowImage**( java.awt.Image si )  
 – Parameters  
 \* si -

---

  - *setSize*  
 public void **setSize**( java.awt.Dimension d )  
 – Parameters  
 \* d -

---

  - *setSize*  
 public void **setSize**( int w, int h )  
 – Parameters  
 \* w -  
 \* h -

#### 16.2.40 CLASS VariableActor

---

**VariableActor** represent graphically the language construct **Variable** for primitive data types and Strings.

##### DECLARATION

---

```
public class VariableActor
extends jeliot.theater.Actor
implements ActorContainer
```

##### CONSTRUCTORS

---

- *VariableActor*  
 public **VariableActor**( )

##### METHODS

---

- *bind*  
 public void **bind**( )

---

- *calcLabelPosition*  
 protected void **calcLabelPosition**( )

---

- *calculateSize*  
 public void **calculateSize**( )

---

- *getName*  
 public String **getName**( )  
 – Returns -

- 
- *getValue*  
`public ValueActor getValue( )`  
 – Returns -

---

  - *paintActor*  
`public void paintActor( java.awt.Graphics g )`
  - *removeActor*  
`public void removeActor( jeliot.theater.Actor actor )`
  - *reserve*  
`public Point reserve( jeliot.theater.ValueActor actor )`  
 – Parameters  
   \* actor -  
 – Returns -

---

  - *setFont*  
`public void setFont( java.awt.Font font )`
  - *setName*  
`public void setName( java.lang.String name )`  
 – Parameters  
   \* name -

---

  - *setSize*  
`public void setSize( int w, int h )`
  - *setValue*  
`public void setValue( jeliot.theater.ValueActor actor )`  
 – Parameters  
   \* actor -

---

  - *setValueColor*  
`public void setValueColor( java.awt.Color valuec )`  
 – Parameters  
   \* valuec -

---

  - *setValueDimension*  
`public void setValueDimension( int w, int h )`  
 – Parameters  
   \* w -  
   \* h -

---

#### METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*  
`public Animation appear( java.awt.Point loc )`  
 – Usage  
   \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.  
 – Parameters  
   \* loc -  
 – Returns -



- 
- *calculateSize*  
`public void calculateSize( )`
  - *clone*  
`public Object clone( )`
  - *fly*  
`public Animation fly( java.awt.Point p )`
    - **Parameters**
      - \* *p* -
    - **Returns** -
- 
- *fly*  
`public Animation fly( java.awt.Point p, int shadow )`
    - **Usage**
      - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
    - **Parameters**
      - \* *p* -
      - \* *shadow* -
    - **Returns** -
- 
- *getActorAt*  
`public Actor getActorAt( int x, int y )`
    - **Parameters**
      - \* *x* -
      - \* *y* -
    - **Returns** -
- 
- *getBackground*  
`public Color getBackground( )`
    - **Returns** -
- 
- *getFont*  
`public Font getFont( )`
    - **Returns** -
- 
- *getFontMetrics*  
`protected FontMetrics getFontMetrics( )`
    - **Returns** -
- 
- *getForeground*  
`public Color getForeground( )`
    - **Returns** -
- 
- *getHeight*  
`public int getHeight( )`
    - **Returns** -
- 
- *getLocation*  
`public Point getLocation( )`
    - **Returns** -
- 
- *getParent*  
`public ActorContainer getParent( )`

- **Returns** -

---

  - *getRootLocation*  
public Point **getRootLocation**( )
  - **Returns** -

---

  - *getShadow*  
public int **getShadow**( )
  - **Returns** -

---

  - *getSize*  
public Dimension **getSize**( )
  - **Returns** -

---

  - *getWidth*  
public int **getWidth**( )
  - **Returns** -

---

  - *getX*  
public int **getX**( )
  - **Returns** -

---

  - *getY*  
public int **getY**( )
  - **Returns** -

---

  - *paintActor*  
public void **paintActor**( java.awt.Graphics g )
  - **Usage**
    - \* Paints the actor on the given Graphics instance. Override this in subclasses.
  - **Parameters**
    - \* g - The Graphics object

---

  - *paintActors*  
protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )
  - **Usage**
    - \* Paints the actors contained in the vector on this actor.
  - **Parameters**
    - \* g -
    - \* actors -

---

  - *paintBackground*  
protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )
  - **Parameters**
    - \* g -
    - \* backImage -
    - \* xx -
    - \* yy -
    - \* w -
    - \* h -

---

  - *paintShadow*  
public void **paintShadow**( java.awt.Graphics g )

- **Usage**
    - \* Paints the shadow of the actor. Override this in the subclasses if needed.
  - **Parameters**
    - \* *g* -

---
- *setBackground*

```
public void setBackground( java.awt.Color bgcolor )
```

  - **Parameters**
    - \* *bgcolor* -

---
- *setBorderWidth*

```
public void setBorderWidth( int w )
```

  - **Parameters**
    - \* *w* -

---
- *setBounds*

```
public void setBounds( int x, int y, int w, int h )
```

  - **Parameters**
    - \* *x* -
    - \* *y* -
    - \* *w* -
    - \* *h* -

---
- *setFont*

```
public void setFont( java.awt.Font font )
```

  - **Parameters**
    - \* *font* -

---
- *setForeground*

```
public void setForeground( java.awt.Color fgcolor )
```

  - **Parameters**
    - \* *fgcolor* -

---
- *setInsets*

```
public void setInsets( java.awt.Insets insets )
```

  - **Parameters**
    - \* *insets* -

---
- *setLight*

```
public void setLight( int light )
```

  - **Parameters**
    - \* *light* -

---
- *setLocation*

```
public void setLocation( int x, int y )
```

  - **Parameters**
    - \* *x* -
    - \* *y* -

---
- *setLocation*

```
public void setLocation( java.awt.Point loc )
```

  - **Parameters**
    - \* *loc* -

---

- *setParent*  
 public void **setParent**( jeliot.theater.ActorContainer parent )  
 — Parameters  
 \* parent -

---

- *setShadow*  
 public void **setShadow**( int s )  
 — Parameters  
 \* s -

---

- *setShadowImage*  
 public static void **setShadowImage**( java.awt.Image si )  
 — Parameters  
 \* si -

---

- *setSize*  
 public void **setSize**( java.awt.Dimension d )  
 — Parameters  
 \* d -

---

- *setSize*  
 public void **setSize**( int w, int h )  
 — Parameters  
 \* w -  
 \* h -

#### 16.2.41 CLASS VariableInArrayActor

---

**VariableInArrayActor** represent graphically the language construct **VariableInArray** for primitive data types and Strings.

##### DECLARATION

---

```
public class VariableInArrayActor
extends jeliot.theater.VariableActor
```

##### FIELDS

---

- private int indexw  
 —

##### CONSTRUCTORS

---

- *VariableInArrayActor*  
 public **VariableInArrayActor**( )  
 — Parameters
- *VariableInArrayActor*  
 public **VariableInArrayActor**( jeliot.theater.ArrayActor arrayActor,  
 java.lang.String name )  
 — Parameters  
 \* arrayActor -  
 \* name -

## METHODS

- 
- *calcLabelPosition*  
protected void calcLabelPosition( )
  - *calculateSize*  
 public void calculateSize( int indexw, int valuew, int h )  
     – Parameters  
         \* indexw -  
         \* valuew -  
         \* h -  


---
  - *paintActor*  
 public void paintActor( java.awt.Graphics g )

## METHODS INHERITED FROM CLASS jeliot.theater.VariableActor

---

( in 16.2.40, page 1318)

- *bind*  
public void bind( )
- *calcLabelPosition*  
protected void calcLabelPosition( )
- *calculateSize*  
public void calculateSize( )
- *getName*  
 public String getName( )  
     – Returns -  


---
- *getValue*  
 public ValueActor getValue( )  
     – Returns -  


---
- *paintActor*  
public void paintActor( java.awt.Graphics g )
- *removeActor*  
public void removeActor( jeliot.theater.Actor actor )
- *reserve*  
 public Point reserve( jeliot.theater.ValueActor actor )  
     – Parameters  
         \* actor -  
     – Returns -  


---
- *setFont*  
public void setFont( java.awt.Font font )
- *setName*  
 public void setName( java.lang.String name )  
     – Parameters  
         \* name -  


---
- *setSize*  
public void setSize( int w, int h )
- *setValue*  
 public void setValue( jeliot.theater.ValueActor actor )

- **Parameters**
  - \* actor -

---

- *setValueColor*

```
public void setValueColor( java.awt.Color valuec )
```

  - **Parameters**
    - \* valuec -

---

- *setValueDimension*

```
public void setValueDimension( int w, int h )
```

  - **Parameters**
    - \* w -
    - \* h -

---

#### METHODS INHERITED FROM CLASS `jeliot.theater.Actor`

---

( in 16.2.2, page 1109)

- *appear*

```
public Animation appear( java.awt.Point loc )
```

  - **Usage**
    - \* Returns an animation that makes the actor appear. Default implementation shows the actor highlighted for given number of milliseconds.
  - **Parameters**
    - \* loc -
  - **Returns -**

---

- *calculateSize*

```
public void calculateSize( )
```

---

- *clone*

```
public Object clone( )
```

---

- *fly*

```
public Animation fly( java.awt.Point p )
```

  - **Parameters**
    - \* p -
  - **Returns -**

---

- *fly*

```
public Animation fly( java.awt.Point p, int shadow )
```

  - **Usage**
    - \* Makes the actor move to given point in given time (millis) Returns a reference to the animation object.
  - **Parameters**
    - \* p -
    - \* shadow -
  - **Returns -**

---

- *getActorAt*

```
public Actor getActorAt( int x, int y )
```

  - **Parameters**
    - \* x -
    - \* y -

- Returns -

---

  - *getBackground*  
public Color **getBackground**( )

– Returns -

---
- *getFont*  
public Font **getFont**( )

– Returns -

---
- *getFontMetrics*  
protected FontMetrics **getFontMetrics**( )

– Returns -

---
- *getForeground*  
public Color **getForeground**( )

– Returns -

---
- *getHeight*  
public int **getHeight**( )

– Returns -

---
- *getLocation*  
public Point **getLocation**( )

– Returns -

---
- *getParent*  
public ActorContainer **getParent**( )

– Returns -

---
- *getRootLocation*  
public Point **getRootLocation**( )

– Returns -

---
- *getShadow*  
public int **getShadow**( )

– Returns -

---
- *getSize*  
public Dimension **getSize**( )

– Returns -

---
- *getWidth*  
public int **getWidth**( )

– Returns -

---
- *getX*  
public int **getX**( )

– Returns -

---
- *getY*  
public int **getY**( )

– Returns -

---

- *paintActor*  
 public void **paintActor**( java.awt.Graphics g )  
 – **Usage**  
   \* Paints the actor on the given Graphics instance. Override this in subclasses.  
 – **Parameters**  
   \* g - The Graphics object  


---
- *paintActors*  
 protected void **paintActors**( java.awt.Graphics g, java.util.Vector actors )  
 – **Usage**  
   \* Paints the actors contained in the vector on this actor.  
 – **Parameters**  
   \* g -  
   \* actors -  


---
- *paintBackground*  
 protected void **paintBackground**( java.awt.Graphics g, java.awt.Image backImage, int xx, int yy, int w, int h )  
 – **Parameters**  
   \* g -  
   \* backImage -  
   \* xx -  
   \* yy -  
   \* w -  
   \* h -  


---
- *paintShadow*  
 public void **paintShadow**( java.awt.Graphics g )  
 – **Usage**  
   \* Paints the shadow of the actor. Override this in the subclasses if needed.  
 – **Parameters**  
   \* g -  


---
- *setBackground*  
 public void **setBackground**( java.awt.Color bgcolor )  
 – **Parameters**  
   \* bgcolor -  


---
- *setBorderWidth*  
 public void **setBorderWidth**( int w )  
 – **Parameters**  
   \* w -  


---
- *setBounds*  
 public void **setBounds**( int x, int y, int w, int h )  
 – **Parameters**  
   \* x -  
   \* y -  
   \* w -  
   \* h -  


---
- *setFont*  
 public void **setFont**( java.awt.Font font )  
 – **Parameters**



- \* font -
- • *setForeground*  
 public void setForeground( java.awt.Color fgcolor )
  - Parameters
  - \* fgcolor -
- • *setInsets*  
 public void setInsets( java.awt.Insets insets )
  - Parameters
  - \* insets -
- • *setLight*  
 public void setLight( int light )
  - Parameters
  - \* light -
- • *setLocation*  
 public void setLocation( int x, int y )
  - Parameters
  - \* x -
  - \* y -
- • *setLocation*  
 public void setLocation( java.awt.Point loc )
  - Parameters
  - \* loc -
- • *setParent*  
 public void setParent( jeliot.theater.ActorContainer parent )
  - Parameters
  - \* parent -
- • *setShadow*  
 public void setShadow( int s )
  - Parameters
  - \* s -
- • *setShadowImage*  
 public static void setShadowImage( java.awt.Image si )
  - Parameters
  - \* si -
- • *setSize*  
 public void setSize( java.awt.Dimension d )
  - Parameters
  - \* d -
- • *setSize*  
 public void setSize( int w, int h )
  - Parameters
  - \* w -
  - \* h -

## Chapter 17

### Package

### koala.dynamicjava.interpreter.error

*Package Contents*

*Page*

---

#### Classes

<b>CatchedExceptionError</b> .....	1330
<i>This error is thrown when a runtime exception appends while interpreting a statement</i>	
<b>ExecutionError</b> .....	1332
<i>This error is thrown when an unexpected error append while interpreting a statement</i>	

---

## 17.1 Classes

### 17.1.1 CLASS *CatchedExceptionError*

This error is thrown when a runtime exception appends while interpreting a statement

#### DECLARATION

```
public class CatchedExceptionError
extends koala.dynamicjava.interpreter.error.ExecutionError
```

#### CONSTRUCTORS

- *CatchedExceptionError*  
 public **CatchedExceptionError**( java.lang.Exception e )  
 – **Usage**  
 \* Constructs an *CatchedExceptionError* with no detail message.

---

- *CatchedExceptionError*  
 public **CatchedExceptionError**( java.lang.Exception e,  
 koala.dynamicjava.tree.Node n )  
 – **Usage**  
 \* Constructs an *CatchedExceptionError* with the specified detail message,  
 filename, line, column and exception.  
 – **Parameters**  
 \* e - the caught exception  
 \* n - the node in the syntax tree where the error occurs

---

- *CatchedExceptionError*  
 public **CatchedExceptionError**( java.lang.String messageString,  
 java.lang.Exception e, koala.dynamicjava.tree.Node n )  
 – **Usage**  
 \* Constructs an *CatchedExceptionError* with the specified detail message,  
 filename, line, column and exception.  
 – **Parameters**  
 \* messageString - the message string that describes the exception  
 \* e - the caught exception  
 \* n - the node in the syntax tree where the error occurs

---

- *CatchedExceptionError*  
 public **CatchedExceptionError**( java.lang.String messageString,  
 java.lang.Throwable t, koala.dynamicjava.tree.Node n )  
 – **Usage**  
 \* Constructs an *CatchedExceptionError* with the specified detail message,  
 filename, line, column and exception.

– **Parameters**

- \* **messageString** - the message string that describes the exception
- \* **t** - the caught exception or error
- \* **n** - the node in the syntax tree where the error occurs

## METHODS

---

- *getException*

**public Throwable getException( )**

– **Usage**

- \* Returns the exception that causes this error throwing

## METHODS INHERITED FROM CLASS `koala.dynamicjava.interpreter.error.ExecutionError`

---

( in 17.1.2, page 1332)

- *getMessage*

**public String getMessage( )**

– **Usage**

- \* Returns the error message string of this exception

---

- *getNode*

**public Node getNode( )**

– **Usage**

- \* Returns the syntax tree node where the error occurs

---

- *printStackTrace*

**public void printStackTrace( )**

– **Usage**

- \* Overridden to delegate to `printStackTrace(PrintStream)` to print nested exception information.

– **See Also**

- \* `koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintStream)`

---

- *printStackTrace*

**public void printStackTrace( java.io.PrintStream s )**

– **Usage**

- \* Overridden to delegate to `printStackTrace(PrintWriter)` to print nested exception information.

– **See Also**

- \* `koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintWriter)`

---

- *printStackTrace*

**public void printStackTrace( java.io.PrintWriter w )**

– **Usage**

- \* Handles all calls to `printStackTrace()`, printing the stack trace of the current exception, and also that of its cause.

METHODS INHERITED FROM CLASS `java.lang.Error`

---

METHODS INHERITED FROM CLASS `java.lang.Throwable`

---

- *fillInStackTrace*  
public synchronized native Throwable fillInStackTrace( )
- *getCause*  
public Throwable getCause( )
- *getLocalizedMessage*  
public String getLocalizedMessage( )
- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

### 17.1.2 CLASS `ExecutionError`

---

This error is thrown when an unexpected error append while interpreting a statement

DECLARATION

---

```
public class ExecutionError
extends java.lang.Error
```

SERIALIZABLE FIELDS

---

- private Node node
  - The syntax tree node where the error occurs
- private String rawMessage
  - The raw message

FIELDS

---

- private static final String BUNDLE
  - The resource bundle name
- public static final String SHOW\_CAUSE\_PROPERTY
  -
- public static final String SHOW\_TRACE\_PROPERTY
  -
- private static final LocalizedMessageReader reader
  - The message reader
- private Node node
  - The syntax tree node where the error occurs
- private String rawMessage
  - The raw message

CONSTRUCTORS

---

- *ExecutionError*  
public **ExecutionError**( )
    - **Usage**
      - \* Constructs an **ExecutionError** with no detail message.
- 
- *ExecutionError*  
public **ExecutionError**( java.lang.String s )
    - **Usage**

\* Constructs an **ExecutionError** with the specified detail message.

– **Parameters**

\* **s** - the detail message (a key in a resource file).

---

• *ExecutionError*

**public ExecutionError( java.lang.String s, koala.dynamicjava.tree.Node n )**

– **Usage**

\* Constructs an **ExecutionError** with the specified detail message, filename, line and column.

– **Parameters**

\* **s** - the detail message (a key in a resource file).

\* **n** - the syntax tree node where the error occurs

---

• *ExecutionError*

**public ExecutionError( java.lang.Throwable thrown )**

## METHODS

---

• *getMessage*

**public String getMessage( )**

– **Usage**

\* Returns the error message string of this exception

---

• *getNode*

**public Node getNode( )**

– **Usage**

\* Returns the syntax tree node where the error occurs

---

• *printStackTrace*

**public void printStackTrace( )**

– **Usage**

\* Overridden to delegate to `printStackTrace(PrintStream)` to print nested exception information.

– **See Also**

\*

`koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintStream)`

---

• *printStackTrace*

**public void printStackTrace( java.io.PrintStream s )**

– **Usage**

\* Overridden to delegate to `printStackTrace(PrintWriter)` to print nested exception information.

– **See Also**

\*  
koala.dynamicjava.interpreter.error.ExecutionError.printStackTrace(PrintWriter)

- *printStackTrace*

**public void printStackTrace( java.io.PrintWriter w )**

- **Usage**

- \* Handles all calls to `printStackTrace()`, printing the stack trace of the current exception, and also that of its cause.

## METHODS INHERITED FROM CLASS `java.lang.Error`

---

## METHODS INHERITED FROM CLASS `java.lang.Throwable`

---

- *fillInStackTrace*  
**public synchronized native Throwable fillInStackTrace( )**
- *getCause*  
**public Throwable getCause( )**
- *getLocalizedMessage*  
**public String getLocalizedMessage( )**
- *getMessage*  
**public String getMessage( )**
- *getOurStackTrace*  
**private synchronized StackTraceElement getOurStackTrace( )**
- *getStackTrace*  
**public StackTraceElement getStackTrace( )**
- *getStackTraceDepth*  
**private native int getStackTraceDepth( )**
- *getStackTraceElement*  
**private native StackTraceElement getStackTraceElement( int )**
- *initCause*  
**public synchronized Throwable initCause( java.lang.Throwable )**
- *printStackTrace*  
**public void printStackTrace( )**
- *printStackTrace*  
**public void printStackTrace( java.io.PrintStream )**
- *printStackTrace*  
**public void printStackTrace( java.io.PrintWriter )**
- *printStackTraceAsCause*  
**private void printStackTraceAsCause( java.io.PrintStream , java.lang.StackTraceElement [] )**
- *printStackTraceAsCause*  
**private void printStackTraceAsCause( java.io.PrintWriter , java.lang.StackTraceElement [] )**
- *setStackTrace*  
**public void setStackTrace( java.lang.StackTraceElement [] )**
- *toString*  
**public String toString( )**
- *writeObject*  
**private synchronized void writeObject( java.io.ObjectOutputStream )**



## Chapter 18

# Package jeliot

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Classes</b>	
<b>FeatureNotImplementedException</b> .....	1337
<i>This exception is thrown when the parser discovers use of a language feature that is not (yet) implemented in Jeliot.</i>	
<b>Jeliot</b> .....	1338
<i>This is the application class of Jeliot 3 that binds together the theatre, the GUI and the DynamicJava Java source interpreter.</i>	
<hr/>	

## 18.1 Classes

### 18.1.1 CLASS *FeatureNotImplementedException*

---

This exception is thrown when the parser discovers use of a language feature that is not (yet) implemented in Jeliot.

#### DECLARATION

---

```
public class FeatureNotImplementedException
extends java.lang.RuntimeException
```

#### CONSTRUCTORS

---

- *FeatureNotImplementedException*  
`public FeatureNotImplementedException( java.lang.String s )`
  - **Usage**
    - \* The only constructor of the *FeatureNoeImplementedException*.
  - **Parameters**
    - \* **s** - The info string of the exception.

#### METHODS

---

- *toString*  
`public String toString( )`
  - **Usage**
    - \* Turns the exception into the string that defines the exception
  - **Returns** - The description of the exception in HTML -form.

#### METHODS INHERITED FROM CLASS *java.lang.RuntimeException*

---

#### METHODS INHERITED FROM CLASS *java.lang.Exception*

---

#### METHODS INHERITED FROM CLASS *java.lang.Throwable*

---

- *fillInStackTrace*  
`public synchronized native Throwable fillInStackTrace( )`
- *getCause*  
`public Throwable getCause( )`

- *getLocalizedMessage*  
public String getLocalizedMessage( )
- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

### 18.1.2 CLASS Jeliot

---

This is the application class of Jeliot 3 that binds together the theatre, the GUI and the DynamicJava Java source interpreter.

#### DECLARATION

---

```
public class Jeliot
extends java.lang.Object
```

FIELDS

---

- private JeliotWindow gui
  - The graphical user interface.
- private Theater theatre
  - Theatre object for showing the animation.
- private ThreadController controller
  - A thread controller object for handling concurrency and starting and pausing the animation.
- private AnimationEngine engine
  - Animation engine to show the animations.
- private CodePane codePane
  - A code pane for showing and highlighting the program source while the program is being animated.
- private Director director
  - A director for animating the program.
- private ImageLoader iLoad
  - An image loader that takes care of loading the required images.

CONSTRUCTORS

---

- *Jeliot*  
 public **Jeliot**( java.lang.String udir )
  - **Usage**
    - \* The only constructor of the Jeliot 3. Loads Theatre theatre -object's background. Initializes JeliotWindow gui -object with parameters this, CodePane codepane, Theatre theatre, AnimationEngine engine, ImageLoader iLoad
  - **Parameters**
    - \* udir -

METHODS

---

- *compile*  
 public void **compile**( )  


---
- *directorFreezed*  
 public void **directorFreezed**( )
  - **Usage**
    - \* Called by director when it freezes to accept input.

---

- *directorPaused*  
**public void directorPaused( )**
  - **Usage**
    - \* Called by director when a step is completed.

---
- *directorResumed*  
**public void directorResumed( )**
  - **Usage**
    - \* Called by director when it resumes from waiting for input.

---
- *main*  
**public static void main( java.lang.String [] args )**
  - **Parameters**
    - \* args -
  - **Exceptions**
    - \* java.io.IOException -

---
- *output*  
**public void output( java.lang.String str )**
  - **Usage**
    - \* Called by the director when user's program outputs a string.
  - **Parameters**
    - \* str - String that is outputted.

---
- *pause*  
**public void pause( )**
  - **Usage**
    - \* Pauses the animation. Called by gui.

---
- *play*  
**public void play( )**
  - **Usage**
    - \* Starts the animation in play mode. Called by gui.

---
- *rewind*  
**public void rewind( )**
  - **Usage**
    - \* Initializes the compiled program to be animated.

---
- *run*  
**public void run( )**
  - **Usage**
    - \* Sets up the user interface.

---

- *runUntil*  
`public void runUntil( int line )`
  - **Parameters**
    - \* `line` -

---
- *runUntilDone*  
`public void runUntilDone( )`

---
- *setSourceCode*  
`public void setSourceCode( java.lang.String sourceCode, java.lang.String methodCall )`
  - **Usage**
    - \* Creates the Lexer and Java11Parser. Compiles a program that is given in a Reader object.
  - **Parameters**
    - \* `sourceCode` - The program source code as a String.
    - \* `methodCall` - The main method call as a String.

---
- *showErrorMessage*  
`public void showErrorMessage( jeliot.mcode.InterpreterError e )`
  - **Parameters**
    - \* `e` -

---
- *showMessagesInDialogs*  
`public boolean showMessagesInDialogs( )`
  - **Returns** -

---
- *step*  
`public void step( )`
  - **Usage**
    - \* Starts the animation in step mode. Called by gui.

## Chapter 19

# Package

## koala.dynamicjava.gui.resource

Package Contents

Page

---

### Interfaces

**ActionMap** ..... 1344

*This interface represents an object which maps actions to strings*

**JComponentModifier** ..... 1344

*This interface must be implemented by actions which need to have an access to their associated component(s)*

### Classes

**ButtonFactory** ..... 1345

*This class represents a button factory which builds buttons from the content of a resource bundle.*

*The resource entries format is (for a button named 'Button'):*

```
Button.text      = text
Button.icon      = icon_name
Button.mnemonic  = mnemonic
Button.action    = action_name
Button.selected  = true | false
Button.tooltip   = tool tip text
```

where

```
text, icon_name and action_name are strings
mnemonic is a character
```

**MenuFactory** ..... 1349

*This class represents a menu factory which builds menubars and menus from the content of a resource file.*

*The resource entries format is (for a menubar named 'MenuBar'):*

MenuBar	= Menu1 Menu2 ...	
<b>MissingListenerException</b>	.....	1355
	<i>Signals a missing listener</i>	
<b>ResourceFormatException</b>	.....	1357
	<i>Signals a format error in a resource bundle</i>	
<b>ResourceManager</b>	.....	1359
	<i>This class offers convenience methods to decode resource bundle entries</i>	
<b>ToolBarFactory</b>	.....	1361
	<i>This class represents a tool bar factory which builds tool bars from the content of a resource file.</i>	

*The resource entries format is (for a tool bar named 'ToolBar'):*

ToolBar	= Item1 Item2 - Item3 ...
---------	---------------------------

---



## 19.1 Interfaces

### 19.1.1 INTERFACE ActionMap

---

This interface represents an object which maps actions to strings

#### DECLARATION

---

```
public interface ActionMap
```

#### METHODS

---

- *getAction*  
`public Action getAction( java.lang.String key )`
  - **Usage**
    - \* Returns the action associated with the given string or null on error
  - **Parameters**
    - \* **key** - the key mapped with the action to get
  - **Exceptions**
    - \* `koala.dynamicjava.gui.resource.MissingListenerException` - if the action is not found

### 19.1.2 INTERFACE JComponentModifier

---

This interface must be implemented by actions which need to have an access to their associated component(s)

#### DECLARATION

---

```
public interface JComponentModifier
```

#### METHODS

---

- *addJComponent*  
`public void addJComponent( javax.swing.JComponent comp )`
  - **Usage**
    - \* Gives a reference to a component to this object
  - **Parameters**
    - \* **comp** - the component associated with this object

## 19.2 Classes

### 19.2.1 CLASS ButtonFactory

---

This class represents a button factory which builds buttons from the content of a resource bundle.

The resource entries format is (for a button named 'Button'):

```

Button.text      = text
Button.icon      = icon_name
Button.mnemonic  = mnemonic
Button.action    = action_name
Button.selected  = true | false
Button.tooltip   = tool tip text
where
  text, icon_name and action_name are strings
  mnemonic is a character

```

#### DECLARATION

---

```

public class ButtonFactory
extends koala.dynamicjava.gui.resource.ResourceManager

```

#### FIELDS

---

- private static final String ICON\_SUFFIX  
—
- private static final String TEXT\_SUFFIX  
—
- private static final String MNEMONIC\_SUFFIX  
—
- private static final String ACTION\_SUFFIX  
—
- private static final String SELECTED\_SUFFIX  
—
- private static final String TOOLTIP\_SUFFIX  
—

- private ActionMap actions
  - The table which contains the actions

## CONSTRUCTORS

---

- *ButtonFactory*

```
public ButtonFactory( java.util.ResourceBundle  rb,
koala.dynamicjava.gui.resource.ActionMap  am )
```

  - **Usage**
    - \* Creates a new button factory
  - **Parameters**
    - \* **rb** - the resource bundle that contains the buttons description.
    - \* **am** - the actions to bind to the button

## METHODS

---

- *createJButton*

```
public JButton createJButton( java.lang.String  name )
```

    - **Usage**
      - \* Creates and returns a new swing button
    - **Parameters**
      - \* **name** - the name of the button in the resource bundle
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a button.  
It is not thrown if the mnemonic and the action keys are missing
      - \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the mnemonic is not a single character
      - \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if the button action is not found in the action map
- 
- *createJCheckBox*

```
public JCheckBox createJCheckBox( java.lang.String  name )
```

    - **Usage**
      - \* Creates and returns a new swing check box
    - **Parameters**
      - \* **name** - the name of the button in the resource bundle
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a button.  
It is not thrown if the mnemonic and the action keys are missing
      - \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the mnemonic is not a single character
      - \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if the button action is not found in the action map

---

- *createJRadioButton*

```
public JRadioButton createJRadioButton( java.lang.String name )
```

- **Usage**

- \* Creates and returns a new swing radio button

- **Parameters**

- \* **name** - the name of the button in the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a button.  
It is not thrown if the mnemonic and the action keys are missing
  - \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the mnemonic is not a single character
  - \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if the button action is not found in the action  
map

---

- *initializeButton*

```
private void initializeButton( javax.swing.AbstractButton b,  
java.lang.String name )
```

- **Usage**

- \* Initializes a button

- **Parameters**

- \* **b** - the button to initialize
  - \* **name** - the button's name

- **Exceptions**

- \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the mnemonic is not a single character
  - \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if the button action is not found in the action map

METHODS INHERITED FROM CLASS `koala.dynamicjava.gui.resource.ResourceManager`

---

( in 19.2.5, page 1359)

- *getBoolean*

```
public boolean getBoolean( java.lang.String key )
```

- **Usage**

- \* Returns the boolean mapped with the given key

- **Parameters**

- \* **key** - a key of the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a resource
  - \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the resource is malformed

---

- *getInteger*

```
public int getInteger( java.lang.String key )
```

- **Usage**

- \* Returns the integer mapped with the given string

- **Parameters**
    - \* **key** - a key of the resource bundle
  - **Exceptions**
    - \* `java.util.MissingResourceException` - if key is not the name of a resource
    - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the resource is malformed
- 
- *getString*  
`public String getString( java.lang.String key )`
    - **Usage**
      - \* Returns the string that is mapped with the given key
    - **Parameters**
      - \* **key** - a key in the resource bundle
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource
- 
- *getStringList*  
`public List getStringList( java.lang.String key )`
    - **Usage**
      - \* Returns the tokens that compose the string mapped with the given key. Delimiters (" " "t" "n" "r" "f") are not returned.
    - **Parameters**
      - \* **key** - a key of the resource bundle
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource
- 
- *getStringList*  
`public List getStringList( java.lang.String key, java.lang.String delim )`
    - **Usage**
      - \* Returns the tokens that compose the string mapped with the given key. Delimiters are not returned.
    - **Parameters**
      - \* **key** - a key of the resource bundle
      - \* **delim** - the delimiters of the tokens
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource
- 
- *getStringList*  
`public List getStringList( java.lang.String key, java.lang.String delim, boolean returnDelims )`
    - **Usage**
      - \* Returns the tokens that compose the string mapped with the given key
    - **Parameters**
      - \* **key** - a key of the resource bundle
      - \* **delim** - the delimiters of the tokens
      - \* **returnDelims** - if true, the delimiters are returned in the list
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource

## 19.2.2 CLASS MenuFactory

---

This class represents a menu factory which builds menubars and menus from the content of a resource file.

The resource entries format is (for a menubar named 'MenuBar'):

```
MenuBar          = Menu1 Menu2 ...

Menu1.type        = RADIO | CHECK | MENU | ITEM
Menu1             = Item1 Item2 - Item3 ...
Menu1.text        = text
Menu1.icon        = icon_name
Menu1.mnemonic    = mnemonic
Menu1.accelerator = accelerator
Menu1.action      = action_name
Menu1.selected    = true | false
...
mnemonic is a single character
accelerator is of the form: mod+mod+...+X
  where mod is Shift, Meta, Alt or Ctrl
'- ' represents a separator
```

All entries are optional except the '.type' entry  
Consecutive RADIO items are put in a ButtonGroup

### DECLARATION

---

```
public class MenuFactory
extends koala.dynamicjava.gui.resource.ResourceManager
```

### FIELDS

---

- private static final String TYPE\_MENU  
—
- private static final String TYPE\_ITEM  
—
- private static final String TYPE\_RADIO  
—
- private static final String TYPE\_CHECK  
—

- private static final String SEPARATOR
  -
- private static final String TYPE\_SUFFIX
  -
- private static final String TEXT\_SUFFIX
  -
- private static final String MNEMONIC\_SUFFIX
  -
- private static final String ACCELERATOR\_SUFFIX
  -
- private static final String ACTION\_SUFFIX
  -
- private static final String SELECTED\_SUFFIX
  -
- private static final String ICON\_SUFFIX
  -
- private ActionMap actions
  - The table which contains the actions
- private ButtonGroup buttonGroup
  - The current radio group

## CONSTRUCTORS

---

- *MenuFactory*

```
public MenuFactory( java.util.ResourceBundle rb,
koala.dynamicjava.gui.resource.ActionMap am )
```

  - **Usage**
    - \* Creates a new menu factory
  - **Parameters**
    - \* **rb** - the resource bundle that contains the menu bar description.
    - \* **am** - the actions to add to menu items

METHODS

---

• *createJCheckBoxMenuItem*

```
public JCheckBoxMenuItem createJCheckBoxMenuItem( java.lang.String  name
)
```

– **Usage**

\* Creates and returns a new swing check box menu item

– **Parameters**

\* **name** - the name of the menu item

– **Exceptions**

- \* `java.util.MissingResourceException` -  
if one of the keys that compose the menu item  
is missing.  
It is not thrown if the mnemonic, the accelerator and the  
action keys are missing
  - \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the mnemonic is not a single character
  - \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if then item action is not found in the action  
map
- 

• *createJMenu*

```
public JMenu createJMenu( java.lang.String  name )
```

– **Usage**

\* Creates and returns a new swing menu

– **Parameters**

\* **name** - the name of the menu bar in the resource bundle

– **Exceptions**

- \* `java.util.MissingResourceException` -  
if one of the keys that compose the menu is  
missing  
It is not thrown if the mnemonic, the accelerator and the  
action keys are missing
  - \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the mnemonic is not a single character
  - \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if a item action is not found in the action map
- 

• *createJMenuBar*

```
public JMenuBar createJMenuBar( java.lang.String  name )
```

– **Usage**

\* Creates and returns a swing menu bar

– **Parameters**

\* **name** - the name of the menu bar in the resource bundle

– **Exceptions**



- \* `java.util.MissingResourceException` -  
if one of the keys that compose the menu is  
missing.  
It is not thrown if the mnemonic, the accelerator and the  
action keys are missing
- \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the mnemonic is not a single character and if  
the accelerator is malformed
- \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if an item action is not found in the action map

---

- *createJMenuComponent*

`protected JComponent createJMenuComponent( java.lang.String name )`

- **Usage**

- \* Creates and returns a menu item or a separator

- **Parameters**

- \* `name` - the name of the menu item or "-" to create a separator

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a menu item.  
It is not thrown if the mnemonic, the accelerator and the  
action keys are missing
- \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
in case of malformed entry
- \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if an item action is not found in the action map

---

- *createJMenuItem*

`public JMenuItem createJMenuItem( java.lang.String name )`

- **Usage**

- \* Creates and returns a new swing menu item

- **Parameters**

- \* `name` - the name of the menu item

- **Exceptions**

- \* `java.util.MissingResourceException` -  
if one of the keys that compose the menu item  
is missing.  
It is not thrown if the mnemonic, the accelerator and the  
action keys are missing
- \* `koala.dynamicjava.gui.resource.ResourceFormatException` -  
if the mnemonic is not a single character
- \* `koala.dynamicjava.gui.resource.MissingListenerException` -  
if then item action is not found in the action  
map

---

- *createJRadioButtonMenuItem*

`public JRadioButtonMenuItem createJRadioButtonMenuItem( java.lang.String name )`

- **Usage**

- \* Creates and returns a new swing radio button menu item
  - **Parameters**
    - \* **name** - the name of the menu item
  - **Exceptions**
    - \* `java.util.MissingResourceException` - if one of the keys that compose the menu item is missing.  
It is not thrown if the mnemonic, the accelerator and the action keys are missing
    - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the mnemonic is not a single character
    - \* `koala.dynamicjava.gui.resource.MissingListenerException` - if then item action is not found in the action map
- 
- *initializeJMenuItem*  
`private void initializeJMenuItem( javax.swing.JMenuItem item, java.lang.String name )`
    - **Usage**
      - \* Initializes a swing menu item
    - **Parameters**
      - \* **item** - the menu item to initialize
      - \* **name** - the name of the menu item
    - **Exceptions**
      - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the mnemonic is not a single character
      - \* `koala.dynamicjava.gui.resource.MissingListenerException` - if then item action is not found in the action map
- 
- *toKeyStroke*  
`protected KeyStroke toKeyStroke( java.lang.String str )`
    - **Usage**
      - \* Translate a string into a key stroke.  
See the class comment for details
    - **Parameters**
      - \* **str** - a string

METHODS INHERITED FROM CLASS `koala.dynamicjava.gui.resource.ResourceManager`

---

( in 19.2.5, page 1359)

- *getBoolean*  
`public boolean getBoolean( java.lang.String key )`
  - **Usage**
    - \* Returns the boolean mapped with the given key
  - **Parameters**
    - \* **key** - a key of the resource bundle

- **Exceptions**
    - \* `java.util.MissingResourceException` - if key is not the name of a resource
    - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the resource is malformed
- 
- *getInteger*

```
public int getInteger( java.lang.String key )
```

    - **Usage**
      - \* Returns the integer mapped with the given string
    - **Parameters**
      - \* `key` - a key of the resource bundle
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource
      - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the resource is malformed
- 
- *getString*

```
public String getString( java.lang.String key )
```

    - **Usage**
      - \* Returns the string that is mapped with the given key
    - **Parameters**
      - \* `key` - a key in the resource bundle
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource
- 
- *getStringList*

```
public List getStringList( java.lang.String key )
```

    - **Usage**
      - \* Returns the tokens that compose the string mapped with the given key. Delimiters (" " "t"n"r"f") are not returned.
    - **Parameters**
      - \* `key` - a key of the resource bundle
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource
- 
- *getStringList*

```
public List getStringList( java.lang.String key, java.lang.String delim )
```

    - **Usage**
      - \* Returns the tokens that compose the string mapped with the given key. Delimiters are not returned.
    - **Parameters**
      - \* `key` - a key of the resource bundle
      - \* `delim` - the delimiters of the tokens
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource
- 
- *getStringList*

```
public List getStringList( java.lang.String key, java.lang.String delim, boolean returnDelims )
```

    - **Usage**
      - \* Returns the tokens that compose the string mapped with the given key
    - **Parameters**

- \* **key** - a key of the resource bundle
- \* **delim** - the delimiters of the tokens
- \* **returnDelims** - if true, the delimiters are returned in the list
- **Exceptions**
  - \* `java.util.MissingResourceException` - if key is not the name of a resource

### 19.2.3 CLASS `MissingListenerException`

---

Signals a missing listener

#### DECLARATION

---

```
public class MissingListenerException
extends java.lang.RuntimeException
```

#### SERIALIZABLE FIELDS

---

- private String `className`
  - The class name of the listener bundle requested
- private String `key`
  - The name of the specific listener requested by the user

#### FIELDS

---

- private String `className`
  - The class name of the listener bundle requested
- private String `key`
  - The name of the specific listener requested by the user

#### CONSTRUCTORS

---

- *MissingListenerException*

```
public MissingListenerException( java.lang.String s, java.lang.String
className, java.lang.String key )
```

  - **Usage**
    - \* Constructs a `MissingListenerException` with the specified information.  
A detail message is a String that describes this particular exception.
  - **Parameters**
    - \* **s** - the detail message
    - \* **classname** - the name of the listener class
    - \* **key** - the key for the missing listener.

METHODS

---

- *getClassName*  
public String **getClassName**( )
  - **Usage**
    - \* Gets parameter passed by constructor.

---

- *getKey*  
public String **getKey**( )
  - **Usage**
    - \* Gets parameter passed by constructor.

---

- *toString*  
public String **toString**( )
  - **Usage**
    - \* Returns a printable representation of this object

METHODS INHERITED FROM CLASS `java.lang.RuntimeException`

---

METHODS INHERITED FROM CLASS `java.lang.Exception`

---

METHODS INHERITED FROM CLASS `java.lang.Throwable`

---

- *fillInStackTrace*  
public synchronized native Throwable **fillInStackTrace**( )
- *getCause*  
public Throwable **getCause**( )
- *getLocalizedMessage*  
public String **getLocalizedMessage**( )
- *getMessage*  
public String **getMessage**( )
- *getOurStackTrace*  
private synchronized StackTraceElement **getOurStackTrace**( )
- *getStackTrace*  
public StackTraceElement **getStackTrace**( )
- *getStackTraceDepth*  
private native int **getStackTraceDepth**( )
- *getStackTraceElement*  
private native StackTraceElement **getStackTraceElement**( int    )
- *initCause*  
public synchronized Throwable **initCause**( java.lang.Throwable    )
- *printStackTrace*  
public void **printStackTrace**( )

- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

#### 19.2.4 CLASS ResourceFormatException

---

Signals a format error in a resource bundle

##### DECLARATION

---

```
public class ResourceFormatException
extends java.lang.RuntimeException
```

##### SERIALIZABLE FIELDS

---

- private String className
  - The class name of the resource bundle requested
- private String key
  - The name of the specific resource requested by the user

##### FIELDS

---

- private String className
  - The class name of the resource bundle requested
- private String key
  - The name of the specific resource requested by the user

## CONSTRUCTORS

---

- *ResourceFormatException*

```
public ResourceFormatException( java.lang.String s, java.lang.String  
    className, java.lang.String key )
```

- **Usage**

- \* Constructs a ResourceFormatException with the specified information.  
A detail message is a String that describes this particular exception.

- **Parameters**

- \* **s** - the detail message
    - \* **classname** - the name of the resource class
    - \* **key** - the key for the malformed resource.

## METHODS

---

- *getClassName*

```
public String getClassName( )
```

- **Usage**

- \* Gets parameter passed by constructor.

- 
- *getKey*

```
public String getKey( )
```

- **Usage**

- \* Gets parameter passed by constructor.

- 
- *toString*

```
public String toString( )
```

- **Usage**

- \* Returns a printable representation of this object

## METHODS INHERITED FROM CLASS java.lang.RuntimeException

---

## METHODS INHERITED FROM CLASS java.lang.Exception

---

## METHODS INHERITED FROM CLASS java.lang.Throwable

---

- *fillInStackTrace*

```
public synchronized native Throwable fillInStackTrace( )
```

- *getCause*

```
public Throwable getCause( )
```

- *getLocalizedMessage*  
public String getLocalizedMessage( )
- *getMessage*  
public String getMessage( )
- *getOurStackTrace*  
private synchronized StackTraceElement getOurStackTrace( )
- *getStackTrace*  
public StackTraceElement getStackTrace( )
- *getStackTraceDepth*  
private native int getStackTraceDepth( )
- *getStackTraceElement*  
private native StackTraceElement getStackTraceElement( int )
- *initCause*  
public synchronized Throwable initCause( java.lang.Throwable )
- *printStackTrace*  
public void printStackTrace( )
- *printStackTrace*  
public void printStackTrace( java.io.PrintStream )
- *printStackTrace*  
public void printStackTrace( java.io.PrintWriter )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintStream ,  
java.lang.StackTraceElement [] )
- *printStackTraceAsCause*  
private void printStackTraceAsCause( java.io.PrintWriter ,  
java.lang.StackTraceElement [] )
- *setStackTrace*  
public void setStackTrace( java.lang.StackTraceElement [] )
- *toString*  
public String toString( )
- *writeObject*  
private synchronized void writeObject( java.io.ObjectOutputStream )

### 19.2.5 CLASS ResourceManager

---

This class offers convenience methods to decode  
resource bundle entries

#### DECLARATION

---

```
public class ResourceManager
extends java.lang.Object
```



## CONSTRUCTORS

---

- *ResourceManager*

```
public ResourceManager( java.util.ResourceBundle rb )
```

- **Usage**

- \* Creates a new resource manager

- **Parameters**

- \* **rb** - a resource bundle

## METHODS

---

- *getBoolean*

```
public boolean getBoolean( java.lang.String key )
```

- **Usage**

- \* Returns the boolean mapped with the given key

- **Parameters**

- \* **key** - a key of the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a resource
  - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the resource is malformed

---

- *getInteger*

```
public int getInteger( java.lang.String key )
```

- **Usage**

- \* Returns the integer mapped with the given string

- **Parameters**

- \* **key** - a key of the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a resource
  - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the resource is malformed

---

- *getString*

```
public String getString( java.lang.String key )
```

- **Usage**

- \* Returns the string that is mapped with the given key

- **Parameters**

- \* **key** - a key in the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a resource

---

- *getStringList*

```
public List getStringList( java.lang.String key )
```

- **Usage**
    - \* Returns the tokens that compose the string mapped with the given key. Delimiters (” “t“n“r“f”) are not returned.
  - **Parameters**
    - \* **key** - a key of the resource bundle
  - **Exceptions**
    - \* `java.util.MissingResourceException` - if key is not the name of a resource
- 
- *getStringList*

```
public List getStringList( java.lang.String key, java.lang.String delim )
```

    - **Usage**
      - \* Returns the tokens that compose the string mapped with the given key. Delimiters are not returned.
    - **Parameters**
      - \* **key** - a key of the resource bundle
      - \* **delim** - the delimiters of the tokens
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource
- 
- *getStringList*

```
public List getStringList( java.lang.String key, java.lang.String delim,
boolean returnDelims )
```

    - **Usage**
      - \* Returns the tokens that compose the string mapped with the given key
    - **Parameters**
      - \* **key** - a key of the resource bundle
      - \* **delim** - the delimiters of the tokens
      - \* **returnDelims** - if true, the delimiters are returned in the list
    - **Exceptions**
      - \* `java.util.MissingResourceException` - if key is not the name of a resource

### 19.2.6 CLASS *ToolBarFactory*

---

This class represents a tool bar factory which builds tool bars from the content of a resource file.

The resource entries format is (for a tool bar named 'ToolBar'):

```
ToolBar          = Item1 Item2 - Item3 ...
See ButtonFactory.java for details about the items
...
'-' represents a separator
```

All entries are optional.

DECLARATION

---

```
public class ToolBarFactory
extends koala.dynamicjava.gui.resource.ResourceManager
```

FIELDS

---

- private static final String SEPARATOR
  -
- private ActionMap actions
  - The table which contains the actions
- private ButtonFactory buttonFactory
  - The button factory

CONSTRUCTORS

---

- *ToolBarFactory*  
public **ToolBarFactory**( java.util.ResourceBundle **rb**,  
koala.dynamicjava.gui.resource.ActionMap **am** )
  - **Usage**
    - \* Creates a new tool bar factory
  - **Parameters**
    - \* **rb** - the resource bundle that contains the menu bar description.
    - \* **am** - the actions to add to menu items

METHODS

---

- *createJButton*  
public JButton **createJButton**( java.lang.String **name** )
  - **Usage**
    - \* Creates and returns a new swing button
  - **Parameters**
    - \* **name** - the name of the button in the resource bundle
  - **Exceptions**
    - \* java.util.MissingResourceException - if key is not the name of a button.  
It is not thrown if the mnemonic and the action keys are missing
    - \* koala.dynamicjava.gui.resource.ResourceFormatException -  
if the mnemonic is not a single character
    - \* koala.dynamicjava.gui.resource.MissingListenerException -  
if the button action is not found in the action map

---

- *createJToolBar*

```
public JToolBar createJToolBar( java.lang.String name )
```

- **Usage**

- \* Creates a tool bar

- **Parameters**

- \* **name** - the name of the menu bar in the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if one of the keys that compose the tool bar is missing.  
It is not thrown if the action key is missing
    - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if a boolean is malformed
    - \* `koala.dynamicjava.gui.resource.MissingListenerException` - if an item action is not found in the action map

METHODS INHERITED FROM CLASS `koala.dynamicjava.gui.resource.ResourceManager`

---

( in 19.2.5, page 1359)

- *getBoolean*

```
public boolean getBoolean( java.lang.String key )
```

- **Usage**

- \* Returns the boolean mapped with the given key

- **Parameters**

- \* **key** - a key of the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a resource
    - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the resource is malformed

---

- *getInteger*

```
public int getInteger( java.lang.String key )
```

- **Usage**

- \* Returns the integer mapped with the given string

- **Parameters**

- \* **key** - a key of the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a resource
    - \* `koala.dynamicjava.gui.resource.ResourceFormatException` - if the resource is malformed

---

- *getString*

```
public String getString( java.lang.String key )
```

- **Usage**

- \* Returns the string that is mapped with the given key

- **Parameters**

- \* **key** - a key in the resource bundle

- **Exceptions**

- \* `java.util.MissingResourceException` - if key is not the name of a resource

---

- *getStringList*

```
public List getStringList( java.lang.String key )
```

  - **Usage**
    - \* Returns the tokens that compose the string mapped with the given key. Delimiters (" " "t" "n" "r" "f") are not returned.
  - **Parameters**
    - \* `key` - a key of the resource bundle
  - **Exceptions**
    - \* `java.util.MissingResourceException` - if key is not the name of a resource

---

- *getStringList*

```
public List getStringList( java.lang.String key, java.lang.String delim )
```

  - **Usage**
    - \* Returns the tokens that compose the string mapped with the given key. Delimiters are not returned.
  - **Parameters**
    - \* `key` - a key of the resource bundle
    - \* `delim` - the delimiters of the tokens
  - **Exceptions**
    - \* `java.util.MissingResourceException` - if key is not the name of a resource

---

- *getStringList*

```
public List getStringList( java.lang.String key, java.lang.String delim, boolean returnDelims )
```

  - **Usage**
    - \* Returns the tokens that compose the string mapped with the given key
  - **Parameters**
    - \* `key` - a key of the resource bundle
    - \* `delim` - the delimiters of the tokens
    - \* `returnDelims` - if true, the delimiters are returned in the list
  - **Exceptions**
    - \* `java.util.MissingResourceException` - if key is not the name of a resource

## Chapter 20

# Package jeliot.io

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Classes</b>	
<b>Input</b> .....	1366
<i>Only a class stub for input handling the actual input handling is         done in koala.dynamicJava.intepreter.EvaluationVisitor and jeliot.mcode.Interpreter.</i>	
<b>Output</b> .....	1367
<i>Only a class stub for output handling the actual output handling is         done in koala.dynamicJava.intepreter.EvaluationVisitor and jeliot.mcode.Interpreter.</i>	
<hr/>	

## 20.1 Classes

### 20.1.1 CLASS Input

---

Only a class stub for input handling the actual input handling is done in `koala.dynamicJava.intepreter.EvaluationVisitor` and `jeliot.mcode.Interpreter`.

#### DECLARATION

---

```
public class Input
extends java.lang.Object
```

#### CONSTRUCTORS

---

- *Input*  
`public Input( )`

#### METHODS

---

- *readChar*  
`public static char readChar( )`
  - **Usage**  
 \* A method stub for character reading.
  - **Returns** - returns value space ' '.

---
- *readDouble*  
`public static double readDouble( )`
  - **Usage**  
 \* A method stub for double reading.
  - **Returns** - returns value zero 0.0.

---
- *readInt*  
`public static int readInt( )`
  - **Usage**  
 \* A method stub for integer reading.
  - **Returns** - returns value zero 0.

---
- *readString*  
`public static String readString( )`
  - **Usage**  
 \* A method stub for string reading.
  - **Returns** - returns value space " ".

### 20.1.2 CLASS Output

---

Only a class stub for output handling the actual output handling is done in `koala.dynamicJava.intepreter.EvaluationVisitor` and `jeliot.mcode.Interpreter`.

#### DECLARATION

---

```
public class Output
extends java.lang.Object
```

#### FIELDS

---

- `private static PrintStream out`
  - A copy of the `System.out` `PrintStream`.

#### CONSTRUCTORS

---

- *Output*  
`public Output( )`

#### METHODS

---

- *println*  
`public static void println( boolean b )`
  - **Usage**
    - \* Prints out the given boolean value.
  - **Parameters**
    - \* `b` - The boolean to be printed.

---
- *println*  
`public static void println( char c )`
  - **Usage**
    - \* Prints out the given character.
  - **Parameters**
    - \* `c` - The character to be printed.

---
- *println*  
`public static void println( double d )`
  - **Usage**
    - \* Prints out the given double value.
  - **Parameters**
    - \* `d` - The double to be printed.



- 
- *println*  
`public static void println( float f )`
    - **Usage**
      - \* Prints out the given float value.
    - **Parameters**
      - \* **f** - The float to be printed.
- 
- *println*  
`public static void println( int i )`
    - **Usage**
      - \* Prints out the given integer value.
    - **Parameters**
      - \* **i** - The integer to be printed.
- 
- *println*  
`public static void println( long l )`
    - **Usage**
      - \* Prints out the given long value.
    - **Parameters**
      - \* **l** - The long to be printed.
- 
- *println*  
`public static void println( java.lang.String s )`
    - **Usage**
      - \* Prints out the given String value.
    - **Parameters**
      - \* **s** - The String to be printed.

## Chapter 21

# Package koala.dynamicjava.tree

<i>Package Contents</i>	<i>Page</i>
<hr/>	
<b>Interfaces</b>	
<b>ContinueTarget</b> .....	1375
<i>The classes that implements this interface can contain a continue statement</i>	
<b>ExpressionContainer</b> .....	1375
<i>This interface is implemented by the classes that contains an expression</i>	
<b>ExpressionStatement</b> .....	1376
<i>This interface represents the statement expressions</i>	
<b>IdentifierToken</b> .....	1376
<i>This interface represents an identifier token, ie a token from the tree point of view</i>	
<b>LeftHandSide</b> .....	1377
<i>This interface represents the target of an assignment</i>	
<b>Classes</b>	
<b>AddAssignExpression</b> .....	1377
<i>This class represents the add assign expression nodes of the syntax tree</i>	
<b>AddExpression</b> .....	1382
<i>This class represents the add expression nodes of the syntax tree</i>	
<b>Allocation</b> .....	1387
<i>This class represents the allocation nodes of the syntax tree</i>	
<b>AndExpression</b> .....	1391
<i>This class represents the and expression nodes of the syntax tree</i>	
<b>ArrayAccess</b> .....	1396
<i>This class represents the array access nodes of the syntax tree</i>	
<b>ArrayAllocation</b> .....	1401
<i>This class represents the array allocation nodes of the syntax tree</i>	
<b>ArrayAllocation.TypeDescriptor</b> .....	1406
<i>This class contains informations about the array to create</i>	
<b>ArrayInitializer</b> .....	1407
<i>This class represents the array initializer nodes of the syntax tree</i>	
<b>ArrayType</b> .....	1412
<i>This class represents the array type nodes of the syntax tree</i>	
<b>AssignExpression</b> .....	1417
<i>This class represents the assign expression nodes of the syntax tree</i>	
<b>BinaryExpression</b> .....	1421

<i>This class represents the binary expression nodes of the syntax tree</i>	
<b>BitAndAssignExpression</b> .....	1426
<i>This class represents the bit and assign expression nodes of the syntax tree</i>	
<b>BitAndExpression</b> .....	1430
<i>This class represents the bit and expression nodes of the syntax tree</i>	
<b>BitOrAssignExpression</b> .....	1435
<i>This class represents the bit and assign expression nodes of the syntax tree</i>	
<b>BitOrExpression</b> .....	1440
<i>This class represents the bit or expression nodes of the syntax tree</i>	
<b>BlockStatement</b> .....	1445
<i>This class represents the block statement nodes of the syntax tree</i>	
<b>BooleanLiteral</b> .....	1449
<i>This class represents the boolean literal nodes of the syntax tree</i>	
<b>BooleanType</b> .....	1454
<i>This class represents the boolean type nodes of the syntax tree</i>	
<b>BreakStatement</b> .....	1459
<i>This class represents the break statement nodes of the syntax tree</i>	
<b>ByteType</b> .....	1463
<i>This class represents the byte type nodes of the syntax tree</i>	
<b>CastExpression</b> .....	1468
<i>This class represents the cast expression nodes of the syntax tree</i>	
<b>CatchStatement</b> .....	1473
<i>To store the informations about the catch statements</i>	
<b>CharacterLiteral</b> .....	1477
<i>This class represents the character literal nodes of the syntax tree</i>	
<b>CharType</b> .....	1482
<i>This class represents the char type nodes of the syntax tree</i>	
<b>ClassAllocation</b> .....	1486
<i>This class represents the anonymous class allocation nodes of the syntax tree</i>	
<b>ClassDeclaration</b> .....	1492
<i>This class represents a class declaration</i>	
<b>ClassInitializer</b> .....	1498
<i>This class represents the class initializer statement nodes of the syntax tree</i>	
<b>ComplementExpression</b> .....	1502
<i>This class represents the complement expression nodes of the syntax tree</i>	
<b>ConditionalExpression</b> .....	1506
<i>This class represents the binary expression nodes of the syntax tree</i>	
<b>ConstructorDeclaration</b> .....	1512
<i>This class represents constructor declarations in an AST</i>	
<b>ConstructorInvocation</b> .....	1518
<i>This class represents the constructor call nodes of the syntax tree</i>	
<b>ContinueStatement</b> .....	1524
<i>This class represents the continue statement nodes of the syntax tree</i>	
<b>DivideAssignExpression</b> .....	1528
<i>This class represents the divide assign expression nodes of the syntax tree</i>	
<b>DivideExpression</b> .....	1533
<i>This class represents the divide expression nodes of the syntax tree</i>	
<b>DoStatement</b> .....	1538
<i>This class represents the do statement nodes of the syntax tree</i>	
<b>DoubleLiteral</b> .....	1544

<i>This class represents the double literal nodes of the syntax tree</i>	
<b>DoubleType</b> .....	1548
<i>This class represents the double type nodes of the syntax tree</i>	
<b>EmptyStatement</b> .....	1553
<i>This class represents the empty statement nodes of the syntax tree</i>	
<b>EqualExpression</b> .....	1557
<i>This class represents the equal expression nodes of the syntax tree</i>	
<b>ExclusiveOrAssignExpression</b> .....	1562
<i>This class represents the exclusive and assign expression nodes of the syntax tree</i>	
<b>ExclusiveOrExpression</b> .....	1566
<i>This class represents the bit or expression nodes of the syntax tree</i>	
<b>Expression</b> .....	1571
<i>This class represents the expression nodes of the syntax tree</i>	
<b>FieldAccess</b> .....	1575
<i>This class represents the field access nodes of the syntax tree</i>	
<b>FieldDeclaration</b> .....	1579
<i>This class represents field declarations in an AST</i>	
<b>FloatLiteral</b> .....	1584
<i>This class represents the float literal nodes of the syntax tree</i>	
<b>FloatType</b> .....	1589
<i>This class represents the float type nodes of the syntax tree</i>	
<b>FormalParameter</b> .....	1594
<i>This class represents the method parameters in the syntax tree</i>	
<b>ForStatement</b> .....	1599
<i>This class represents the for statement nodes of the syntax tree</i>	
<b>FunctionCall</b> .....	1605
<i>This class represents the function call nodes of the syntax tree</i>	
<b>GreaterExpression</b> .....	1610
<i>This class represents the greater expression nodes of the syntax tree</i>	
<b>GreaterOrEqualExpression</b> .....	1615
<i>This class represents the greater or equal expression nodes of the syntax tree</i>	
<b>Identifier</b> .....	1620
<i>This class implements a tree token</i>	
<b>IfThenElseStatement</b> .....	1621
<i>This class represents the if-then-else statement nodes of the syntax tree</i>	
<b>IfThenStatement</b> .....	1627
<i>This class represents the if-then statement nodes of the syntax tree</i>	
<b>ImportDeclaration</b> .....	1632
<i>This class represents the import declarations</i>	
<b>Initializer</b> .....	1637
<i>This class represents the initializer statement nodes of the syntax tree</i>	
<b>InnerAllocation</b> .....	1641
<i>This class represents the inner allocation nodes of the syntax tree</i>	
<b>InnerClassAllocation</b> .....	1646
<i>This class represents the anonymous inner classe allocation nodes of the syntax tree</i>	
<b>InstanceInitializer</b> .....	1652
<i>This class represents the instance initializer statement nodes of the syntax tree</i>	
<b>InstanceOfExpression</b> .....	1656
<i>This class represents the instanceof expression nodes of the syntax tree</i>	

<b>IntegerLiteral</b> .....	1661
<i>This class represents the integer literal nodes of the syntax tree</i>	
<b>InterfaceDeclaration</b> .....	1667
<i>This class represents an interface declaration</i>	
<b>IntType</b> .....	1672
<i>This class represents the int type nodes of the syntax tree</i>	
<b>LabeledStatement</b> .....	1676
<i>This class represents the labeled statement nodes of the syntax tree</i>	
<b>LessExpression</b> .....	1681
<i>This class represents the less expression nodes of the syntax tree</i>	
<b>LessOrEqualExpression</b> .....	1686
<i>This class represents the less or equal expression nodes of the syntax tree</i>	
<b>Literal</b> .....	1691
<i>This class represents the literal nodes of the syntax tree</i>	
<b>LongLiteral</b> .....	1696
<i>This class represents the long literal nodes of the syntax tree</i>	
<b>LongType</b> .....	1702
<i>This class represents the long type nodes of the syntax tree</i>	
<b>MethodCall</b> .....	1706
<i>This class represents the method call nodes of the syntax tree</i>	
<b>MethodDeclaration</b> .....	1711
<i>This class represents method declarations in an AST</i>	
<b>MinusExpression</b> .....	1717
<i>This class represents the minus expression nodes of the syntax tree</i>	
<b>MultiplyAssignExpression</b> .....	1722
<i>This class represents the multiply assign expression nodes of the syntax tree</i>	
<b>MultiplyExpression</b> .....	1727
<i>This class represents the multiply expression nodes of the syntax tree</i>	
<b>Node</b> .....	1732
<i>This class represents the nodes of the syntax tree</i>	
<b>NotEqualExpression</b> .....	1736
<i>This class represents the not equal expression nodes of the syntax tree</i>	
<b>NotExpression</b> .....	1741
<i>This class represents the not expression nodes of the syntax tree</i>	
<b>NullLiteral</b> .....	1745
<i>This class represents the null literal nodes of the syntax tree</i>	
<b>ObjectFieldAccess</b> .....	1750
<i>This class represents the field access nodes of the syntax tree</i>	
<b>ObjectMethodCall</b> .....	1755
<i>This class represents the method call nodes of the syntax tree</i>	
<b>OrExpression</b> .....	1760
<i>This class represents the or expression nodes of the syntax tree</i>	
<b>PackageDeclaration</b> .....	1765
<i>This class represents the package declarations</i>	
<b>PlusExpression</b> .....	1770
<i>This class represents the plus expression nodes of the syntax tree</i>	
<b>PostDecrement</b> .....	1774
<i>This class represents the post decrement nodes of the syntax tree</i>	
<b>PostIncrement</b> .....	1778
<i>This class represents the post increment nodes of the syntax tree</i>	

<b>PreDecrement</b> .....	1783
<i>This class represents the pre decrement nodes of the syntax tree</i>	
<b>PreIncrement</b> .....	1787
<i>This class represents the pre increment nodes of the syntax tree</i>	
<b>PrimaryExpression</b> .....	1792
<i>This class represents the primary expression nodes of the syntax tree</i>	
<b>PrimitiveType</b> .....	1795
<i>This class represents the primitive type nodes of the syntax tree</i>	
<b>QualifiedName</b> .....	1800
<i>This class represents the qualified name nodes of the syntax tree</i>	
<b>ReferenceType</b> .....	1805
<i>This class represents the reference type nodes of the syntax tree</i>	
<b>RemainderAssignExpression</b> .....	1810
<i>This class represents the remainder assign expression nodes of the syntax tree</i>	
<b>RemainderExpression</b> .....	1815
<i>This class represents the remainder expression nodes of the syntax tree</i>	
<b>ReturnStatement</b> .....	1820
<i>This class represents the return statement nodes of the syntax tree</i>	
<b>ShiftLeftAssignExpression</b> .....	1824
<i>This class represents the shift left assign expression nodes of the syntax tree</i>	
<b>ShiftLeftExpression</b> .....	1829
<i>This class represents the shift left expression nodes of the syntax tree</i>	
<b>ShiftRightAssignExpression</b> .....	1834
<i>This class represents the shift right assign expression nodes of the syntax tree</i>	
<b>ShiftRightExpression</b> .....	1838
<i>This class represents the shift right expression nodes of the syntax tree</i>	
<b>ShortType</b> .....	1843
<i>This class represents the short type nodes of the syntax tree</i>	
<b>SimpleAllocation</b> .....	1847
<i>This class represents the allocation nodes of the syntax tree</i>	
<b>SimpleAssignExpression</b> .....	1852
<i>This class represents the assign expression nodes of the syntax tree</i>	
<b>Statement</b> .....	1857
<i>This class represents the statement nodes of the syntax tree</i>	
<b>StaticFieldAccess</b> .....	1861
<i>This class represents the field access nodes of the syntax tree</i>	
<b>StaticMethodCall</b> .....	1866
<i>This class represents the static method call nodes of the syntax tree</i>	
<b>StringLiteral</b> .....	1871
<i>This class represents the string literal nodes of the syntax tree</i>	
<b>SubtractAssignExpression</b> .....	1876
<i>This class represents the subtract assign expression nodes of the syntax tree</i>	
<b>SubtractExpression</b> .....	1881
<i>This class represents the subtract expression nodes of the syntax tree</i>	
<b>SuperFieldAccess</b> .....	1886
<i>This class represents the super field access nodes of the syntax tree</i>	
<b>SuperMethodCall</b> .....	1890
<i>This class represents the super method call nodes of the syntax tree</i>	
<b>SwitchBlock</b> .....	1895
<i>This class represents the switch expression-statement bindings</i>	

<b>SwitchStatement</b> .....	1900
<i>This class represents the switch statement nodes of the syntax tree</i>	
<b>SynchronizedStatement</b> .....	1905
<i>This class represents the synchronized statement nodes of the syntax tree</i>	
<b>ThisExpression</b> .....	1910
<i>This class represents the 'this' expression nodes of the syntax tree</i>	
<b>ThrowStatement</b> .....	1914
<i>This class represents the throw statement nodes of the syntax tree</i>	
<b>TreeUtilities</b> .....	1919
<i>This class contains a collection of utility methods for trees.</i>	
<b>TryStatement</b> .....	1920
<i>This class represents the try statement nodes of the syntax tree</i>	
<b>Type</b> .....	1924
<i>This class represents the type nodes of the syntax tree</i>	
<b>TypeDeclaration</b> .....	1928
<i>This class represents a type declaration</i>	
<b>TypeExpression</b> .....	1933
<i>This class represents the type expression nodes of the syntax tree</i>	
<b>UnaryExpression</b> .....	1938
<i>This class represents the unary expression nodes of the syntax tree</i>	
<b>UnsignedShiftRightAssignExpression</b> .....	1942
<i>This class represents the unsigned shift right assign expression nodes of the syntax tree</i>	
<b>UnsignedShiftRightExpression</b> .....	1947
<i>This class represents the unsigned shift right expression nodes of the syntax tree</i>	
<b>VariableDeclaration</b> .....	1952
<i>This class represents variable declarations in an AST</i>	
<b>VoidType</b> .....	1958
<i>This class represents the void type nodes of the syntax tree</i>	
<b>WhileStatement</b> .....	1962
<i>This class represents the while statement nodes of the syntax tree</i>	

---

## 21.1 Interfaces

### 21.1.1 INTERFACE ContinueTarget

---

The classes that implements this interface can contain a continue statement

#### DECLARATION

---

```
public interface ContinueTarget
```

#### METHODS

---

- *addLabel*  
public void **addLabel**( java.lang.String label )
  - **Usage**
    - \* Adds a label to this statement
  - **Parameters**
    - \* label - the label to add

---

- *hasLabel*  
public boolean **hasLabel**( java.lang.String label )
  - **Usage**
    - \* Test whether this statement has the given label
  - **Returns** - true if this statement has the given label

### 21.1.2 INTERFACE ExpressionContainer

---

This interface is implemented by the classes that contains an expression

#### DECLARATION

---

```
public interface ExpressionContainer
```

#### FIELDS

---

- public static final String **EXPRESSION**
  - The expression property name



## METHODS

- *getExpression*  
 public Expression **getExpression**( )  
 – **Usage**  
 \* Returns the expression

---

- *setExpression*  
 public void **setExpression**( koala.dynamicjava.tree.Expression e )  
 – **Usage**  
 \* Sets the expression

## 21.1.3 INTERFACE ExpressionStatement

This interface represents the statement expressions

## DECLARATION

```
public interface ExpressionStatement
```

## 21.1.4 INTERFACE IdentifierToken

This interface represents an identifier token, ie  
 a token from the tree point of view

## DECLARATION

```
public interface IdentifierToken
```

## METHODS

- *beginColumn*  
 public int **beginColumn**( )  
 – **Usage**  
 \* Returns the column number where the beginning of the token  
 was found in the source file

---

- *beginLine*  
 public int **beginLine**( )  
 – **Usage**

- \* Returns the line number where the beginning of the token was found in the source file

---

- *endColumn*

```
public int endColumn( )
```

- **Usage**

- \* Returns the column number where the end of the token was found in the source file

---

- *endLine*

```
public int endLine( )
```

- **Usage**

- \* Returns the line number where the end of the token was found in the source file

---

- *image*

```
public String image( )
```

- **Usage**

- \* Returns the representation of the identifier

### 21.1.5 INTERFACE **LeftHandSide**

---

This interface represents the target of an assignment

DECLARATION

---

```
public interface LeftHandSide
```

## 21.2 Classes

### 21.2.1 CLASS **AddAssignExpression**

---

This class represents the add assign expression nodes of the syntax tree

DECLARATION

---

```
public class AddAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

CONSTRUCTORS

---

• *AddAssignExpression*

```
public AddAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

• *AddAssignExpression*

```
public AddAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

\* **fn** - the filename

\* **bl** - the begin line

\* **bc** - the begin column

\* **el** - the end line

\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---

( in 21.2.10, page 1417)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**
    - \* Returns the left hand side expression
- 
- *getRightExpression*  
 public Expression **getRightExpression**( )
    - **Usage**
      - \* Returns the right hand side expression
- 
- *setLeftExpression*  
 public void **setLeftExpression**( koala.dynamicjava.tree.Expression exp )
    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*  
 public void **setRightExpression**( koala.dynamicjava.tree.Expression exp )
    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
    - **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
- **Parameters**
  - \* **propertyName** - The name of the property to listen on.
  - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, boolean **oldValue**, boolean **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
public int **getBeginColumn**( )
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
public int **getBeginLine**( )
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
public int **getEndColumn**( )
  - **Usage**
    - \* Returns the end column of this node in the end line

- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )

- **Usage**
  - \* Sets the begin column
- - *setBeginLine*
  - public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line
- - *setEndColumn*
  - public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column
- - *setEndLine*
  - public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line
- - *setFilename*
  - public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename
- - *setProperty*
  - public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
    - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.2 CLASS AddExpression

---

This class represents the add expression nodes of the syntax tree

#### DECLARATION

---

```
public class AddExpression
extends koala.dynamicjava.tree.BinaryExpression
```

#### CONSTRUCTORS

---

- *AddExpression*
  - public **AddExpression**( koala.dynamicjava.tree.Expression lexp,  
koala.dynamicjava.tree.Expression rexp )
  - **Usage**
    - \* Initializes the expression
    - **Parameters**

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression

---

- *AddExpression*

```
public AddExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

---

## METHODS

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

( in 21.2.11, page 1421)

- *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**

- \* Returns the left hand side expression

---

- *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**

- \* Returns the right hand side expression

---

- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

- **Usage**

- \* Sets the left hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null
-



- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

- **Usage**

- \* Sets the right hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

- **Parameters**

- \* `propertyName` - The name of the property to listen on.
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

- **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.

- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code

---

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.

---

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

- **Usage**
  - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.3 CLASS Allocation

---

This class represents the allocation nodes of the syntax tree

#### DECLARATION

---

```
public abstract class Allocation
extends koala.dynamicjava.tree.PrimaryExpression
```

#### FIELDS

---

- public static final String CREATION\_TYPE
  - The creationType property name
- private Type creationType
  - The creationType

#### CONSTRUCTORS

---

- *Allocation*

```
protected Allocation( koala.dynamicjava.tree.Type tp, java.lang.String fn,
int bl, int bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **tp** - the creation type
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

- 
- *getCreationType*  
 public Type **getCreationType**( )  
 – **Usage**  
   \* Returns the creation type
- 
- *setCreationType*  
 public void **setCreationType**( koala.dynamicjava.tree.Type t )  
 – **Usage**  
   \* Sets the creation type  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if t is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* **visitor** - the visitor to accept
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener to the listener list.  
   The listener is registered for all properties.  
 – **Parameters**  
   \* **listener** - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**

- \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

– **Parameters**

- \* `propertyName` - The name of the property to listen on.
- \* `listener` - The `PropertyChangeListener` to be added

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )

- **Usage**
  - \* Sets the begin column
- - *setBeginLine*
  - public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line
- - *setEndColumn*
  - public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column
- - *setEndLine*
  - public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line
- - *setFilename*
  - public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename
- - *setProperty*
  - public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
    - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.4 CLASS AndExpression

---

This class represents the and expression nodes of the syntax tree

### DECLARATION

---

```
public class AndExpression
extends koala.dynamicjava.tree.BinaryExpression
```

### CONSTRUCTORS

---

- *AndExpression*
  - public **AndExpression**( koala.dynamicjava.tree.Expression lexp,  
koala.dynamicjava.tree.Expression rexp )
  - **Usage**
    - \* Initializes the expression
    - **Parameters**



- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression

---

- *AndExpression*

```
public AndExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**

- \* Returns the left hand side expression

---

- *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**

- \* Returns the right hand side expression

---

- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

- **Usage**

- \* Sets the left hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null
-

- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

- **Usage**

- \* Sets the right hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

- **Parameters**

- \* `propertyName` - The name of the property to listen on.
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

- **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.

- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code

---

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.

---

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

- **Usage**
    - \* Sets the end line
- 
- *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.5 CLASS ArrayAccess

This class represents the array access nodes of the syntax tree

#### DECLARATION

```
public class ArrayAccess
extends koala.dynamicjava.tree.PrimaryExpression
implements LeftHandSide, ExpressionContainer
```

#### FIELDS

- public static final String EXPRESSION
  - The expression property name
- public static final String CELL\_NUMBER
  - The cellNumber property name
- private Expression expression
  - The expression on which this array access applies
- private Expression cellNumber
  - The expression which denotes the cell number

CONSTRUCTORS

---

• *ArrayAccess*

```
public ArrayAccess( koala.dynamicjava.tree.Expression  exp,
koala.dynamicjava.tree.Expression  cell )
```

– **Usage**

\* Creates a new array access node

– **Parameters**

\* **exp** - the expression on which this array access applies  
 \* **cell** - the cell number

---

• *ArrayAccess*

```
public ArrayAccess( koala.dynamicjava.tree.Expression  exp,
koala.dynamicjava.tree.Expression  cell, java.lang.String  fn, int  bl, int
bc, int  el, int  ec )
```

– **Usage**

\* Creates a new array access node

– **Parameters**

\* **exp** - the expression on which this array access applies  
 \* **cell** - the cell number  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getCellNumber*

```
public Expression getCellNumber( )
```

– **Usage**

\* Returns the expression which denotes the cell number

---

• *getExpression*

```
public Expression getExpression( )
```

– **Usage**

\* Returns the expression on which this array access applies

---

- *setCellNumber*

```
public void setCellNumber( koala.dynamicjava.tree.Expression e )
```

- **Usage**

- \* Sets the expression which denotes the cell number

- **Exceptions**

- \* java.lang.IllegalArgumentException - if e is null

---

- *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression e )
```

- **Usage**

- \* Sets the expression on which this array access applies

- **Exceptions**

- \* java.lang.IllegalArgumentException - if e is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* visitor - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* listener - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

– **Parameters**

- \* `propertyName` - The name of the property to listen on.
- \* `listener` - The `PropertyChangeListener` to be added

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line



- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )

- **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.6 CLASS ArrayAllocation

---

This class represents the array allocation nodes of the syntax tree

#### DECLARATION

---

```
public class ArrayAllocation
extends koala.dynamicjava.tree.Allocation
```

#### FIELDS

---

- private ArrayAllocation.TypeDescriptor typeDescriptor
  - The type descriptor

CONSTRUCTORS

---

• *ArrayAllocation*

```
public ArrayAllocation( koala.dynamicjava.tree.Type tp,
koala.dynamicjava.tree.ArrayAllocation.TypeDescriptor td )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **tp** - the type prefix  
 \* **td** - the type descriptor

---

• *ArrayAllocation*

```
public ArrayAllocation( koala.dynamicjava.tree.Type tp,
koala.dynamicjava.tree.ArrayAllocation.TypeDescriptor td, java.lang.String
fn, int bl, int bc, int el, int ec )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **tp** - the type prefix  
 \* **td** - the type descriptor  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getDimension*

```
public int getDimension( )
```

– **Usage**

\* Returns the dimension of the array

---

• *getInitialization*

```
public ArrayInitializer getInitialization( )
```

– **Usage**

\* Returns the initialization expression

- 
- *getSizes*  
`public List getSizes( )`
    - **Usage**  
\* Returns the size expressions

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Allocation`

---

( in 21.2.3, page 1387)

- *getCreationType*  
`public Type getCreationType( )`
  - **Usage**  
\* Returns the creation type
- *setCreationType*  
`public void setCreationType( koala.dynamicjava.tree.Type t )`
  - **Usage**  
\* Sets the creation type
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if `t` is null

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**  
\* Allows a visitor to traverse the tree
  - **Parameters**  
\* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`
  - **Usage**  
\* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.

- **Parameters**
    - \* *listener* - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* *propertyName* - The name of the property to listen on.
    - \* *listener* - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
    - **Usage**
      - \* Returns the end column of this node in the end line

---

- *getEndLine*  
**public int getEndLine( )**
    - **Usage**
      - \* Returns the end line of this node in the source code

---

- *getFilename*  
**public String getFilename( )**
    - **Usage**
      - \* Returns the filename. Can be null.

---

- *getProperties*  
**public Set getProperties( )**
    - **Usage**
      - \* Returns the defined properties for this node.
    - **Returns** - a set of string

---

- *getProperty*  
**public Object getProperty( java.lang.String name )**
    - **Usage**
      - \* Returns the value of a property
    - **Parameters**
      - \* **name** - the property name
    - **Returns** - null if the property was not previously set

---

- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* **name** - the name of the property

---

- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
    - **Usage**

- \* Removes a PropertyChangeListener for a specific property.
- **Parameters**
  - \* **propertyName** - The name of the property that was listened on.
  - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.7 CLASS ArrayAllocation.TypeDescriptor

---

This class contains informations about the array to create

#### DECLARATION

---

```
public static class ArrayAllocation.TypeDescriptor
extends java.lang.Object
```

FIELDS

---

- public int endLine
  - The end line
- public int endColumn
  - The end column

CONSTRUCTORS

---

- *ArrayAllocation.TypeDescriptor*  
 public **ArrayAllocation.TypeDescriptor**( java.util.List sizes, int dim,  
 koala.dynamicjava.tree.ArrayInitializer init, int el, int ec )  
 – **Usage**  
 \* Creates a new type descriptor

METHODS

---

- *initialize*  
 void **initialize**( koala.dynamicjava.tree.Type t )  
 – **Usage**  
 \* Initializes the type descriptor

**21.2.8 CLASS ArrayInitializer**

---

This class represents the array initializer nodes of the syntax tree

DECLARATION

---

```
public class ArrayInitializer
extends koala.dynamicjava.tree.Expression
```

FIELDS

---

- public static final String CELLS
  - The cells property name
- public static final String ELEMENT\_TYPE
  - The element type property name
- private List cells
  - The list of initialized cells
- private Type elementType
  - The element type



CONSTRUCTORS

---

• *ArrayInitializer*

```
public ArrayInitializer( java.util.List  cells )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **cells** - the list of initialized cells

---

• *ArrayInitializer*

```
public ArrayInitializer( java.util.List  cells, java.lang.String  fn, int  bl,
int  bc, int  el, int  ec )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **cells** - the list of initialized cells  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getCells*

```
public List  getCells( )
```

– **Usage**

\* Returns the list of cell initialization expressions

---

• *getElementType*

```
public Type  getElementType( )
```

– **Usage**

\* Returns the element type

– **Exceptions**

\* `java.lang.IllegalStateException` - if elementType is null

---

- *setCells*

```
public void setCells( java.util.List l )
```

- **Usage**

- \* Sets the list of cell initialization expressions

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if `l` is null

---

- *setElementType*

```
public void setElementType( koala.dynamicjava.tree.Type t )
```

- **Usage**

- \* Sets the element type

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if `t` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

- **Parameters**

- \* `propertyName` - The name of the property to listen on.
  - \* `listener` - The `PropertyChangeListener` to be added

- 
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, boolean **oldValue**, boolean **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

- *getFilename*  
**public String getFilename( )**  
  - **Usage**  
 \* Returns the filename. Can be null.

---
- *getProperties*  
**public Set getProperties( )**  
  - **Usage**  
 \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns the value of a property
  - **Parameters**  
 \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns true if a property is defined for this node
  - **Parameters**  
 \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**  
 \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**  
  - **Usage**  
 \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**

- **Usage**
    - \* Sets the begin line
- - *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
- - *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- - *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- - *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.9 CLASS ArrayType

---

This class represents the array type nodes of the syntax tree

#### DECLARATION

---

```
public class ArrayType
extends koala.dynamicjava.tree.Type
```

#### FIELDS

---

- public static final String ELEMENT\_TYPE
  - The elementType property name
- private Type elementType
  - The type of the elements of the arrays represented by this type

CONSTRUCTORS

---

• *ArrayType*

```
public ArrayType( koala.dynamicjava.tree.Type et, int dim )
```

– **Usage**

- \* Initializes the type

– **Parameters**

- \* **et** - the element type

- \* **dim** - the dimension of the arrays represented by this type (> 0)

---

• *ArrayType*

```
public ArrayType( koala.dynamicjava.tree.Type et, int dim,  
java.lang.String fn, int bl, int bc, int el, int ec )
```

– **Usage**

- \* Initializes the type

– **Parameters**

- \* **et** - the element type

- \* **dim** - the dimension of the arrays represented by this type (> 0)

- \* **fn** - the filename

- \* **bl** - the begin line

- \* **bc** - the begin column

- \* **el** - the end line

- \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

- \* Allows a visitor to traverse the tree

– **Parameters**

- \* **visitor** - the visitor to accept

---

• *getElementType*

```
public Type getElementType( )
```

– **Usage**

- \* Returns the type of the elements of the arrays represented by this type

---

• *setElementType*

```
public void setElementType( koala.dynamicjava.tree.Type t )
```

– **Usage**

- \* Sets the type of the elements of the arrays represented by this type

– **Exceptions**

- \* `java.lang.IllegalArgumentException` - if **t** is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property



- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String **name** )  
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )  
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**, java.beans.PropertyChangeListener **listener** )  
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
 public void **setEndLine**( int **i** )  
  - **Usage**
    - \* Sets the end line

---
- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
  - **Usage**

- \* Sets the filename
- 
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.10 CLASS AssignExpression

---

This class represents the assign expression nodes of the syntax tree

#### DECLARATION

---

```
public abstract class AssignExpression
extends koala.dynamicjava.tree.BinaryExpression
implements ExpressionStatement
```

#### CONSTRUCTORS

---

- *AssignExpression*  
 protected **AssignExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int  
 bc, int el, int ec )
  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **lexp** - the LHS expression
    - \* **rexp** - the RHS expression
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*  
 public Expression **getLeftExpression**( )
    - **Usage**
      - \* Returns the left hand side expression
- 
- *getRightExpression*  
 public Expression **getRightExpression**( )

- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.

\* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code

- 
- *getFilename*  
`public String getFilename( )`  
    - **Usage**  
 \* Returns the filename. Can be null.
- 
- *getProperties*  
`public Set getProperties( )`  
    - **Usage**  
 \* Returns the defined properties for this node.
    - **Returns** - a set of string
- 
- *getProperty*  
`public Object getProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns the value of a property
    - **Parameters**  
 \* `name` - the property name
    - **Returns** - null if the property was not previously set
- 
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns true if a property is defined for this node
    - **Parameters**  
 \* `name` - the name of the property
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**  
 \* `listener` - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**  
 \* `propertyName` - The name of the property that was listened on.  
 \* `listener` - The PropertyChangeListener to be removed
- 
- *setBeginColumn*  
`public void setBeginColumn( int i )`  
    - **Usage**  
 \* Sets the begin column
- 
- *setBeginLine*  
`public void setBeginLine( int i )`

- **Usage**
  - \* Sets the begin line
- *setEndColumn*


---

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*


---

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*


---

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*


---

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.11 CLASS BinaryExpression

---

This class represents the binary expression nodes of the syntax tree

#### DECLARATION

---

```
public abstract class BinaryExpression
extends koala.dynamicjava.tree.Expression
```

#### FIELDS

---

- public static final String LEFT\_EXPRESSION
  - The leftExpression property name
- public static final String RIGHT\_EXPRESSION
  - The rightExpression property name
- private Expression leftExpression
  - The LHS expression
- private Expression rightExpression
  - The RHS expression

CONSTRUCTORS

---

• *BinaryExpression*

```
protected BinaryExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

– **Usage**

\* Returns the left hand side expression

• *getRightExpression*

```
public Expression getRightExpression( )
```

– **Usage**

\* Returns the right hand side expression

• *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

– **Usage**

\* Sets the left hand side expression

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if exp is null

• *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

– **Usage**

\* Sets the right hand side expression

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.



- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- - *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column
- - *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line
- - *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column
- - *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line
- - *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename
- - *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.12 CLASS BitAndAssignExpression

---

This class represents the bit and assign expression nodes of the syntax tree

#### DECLARATION

---

```
public class BitAndAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

#### CONSTRUCTORS

---

- *BitAndAssignExpression*  
 public **BitAndAssignExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp )
  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* lexp - the LHS expression
    - \* rexp - the RHS expression

---

- *BitAndAssignExpression*  
 public **BitAndAssignExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int  
 bc, int el, int ec )
  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* lexp - the LHS expression
    - \* rexp - the RHS expression
    - \* fn - the filename
    - \* bl - the begin line
    - \* bc - the begin column
    - \* el - the end line
    - \* ec - the end column

#### METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* visitor - the visitor to accept

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.AssignExpression`

---

( in 21.2.10, page 1417)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.BinaryExpression`

---

( in 21.2.11, page 1421)

- *getLeftExpression*  
`public Expression getLeftExpression( )`  
  - **Usage**  
 \* Returns the left hand side expression
- *getRightExpression*  
`public Expression getRightExpression( )`  
  - **Usage**  
 \* Returns the right hand side expression
- *setLeftExpression*  
`public void setLeftExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
 \* Sets the left hand side expression
  - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if exp is null
- *setRightExpression*  
`public void setRightExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
 \* Sets the right hand side expression
  - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
  - **Usage**  
 \* Allows a visitor to traverse the tree
  - **Parameters**  
 \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`

- **Usage**
  - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
- **Parameters**
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

- 
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
 \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
 \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
 \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
 \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
 \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
 \* Sets the value of a property  
 – **Parameters**  
 \* **name** - the property name  
 \* **value** - the new value to set

### 21.2.13 CLASS BitAndExpression

---

This class represents the bit and expression nodes of the syntax tree

#### DECLARATION

---

```
public class BitAndExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *BitAndExpression*

```
public BitAndExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression

---

• *BitAndExpression*

```
public BitAndExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

## – Usage

\* Returns the left hand side expression

---

• *getRightExpression*

```
public Expression getRightExpression( )
```



- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.

---

\* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code

- 
- *getFilename*  
`public String getFilename( )`  
    - **Usage**  
 \* Returns the filename. Can be null.
- 
- *getProperties*  
`public Set getProperties( )`  
    - **Usage**  
 \* Returns the defined properties for this node.
    - **Returns** - a set of string
- 
- *getProperty*  
`public Object getProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns the value of a property
    - **Parameters**  
 \* `name` - the property name
    - **Returns** - null if the property was not previously set
- 
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns true if a property is defined for this node
    - **Parameters**  
 \* `name` - the name of the property
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**  
 \* `listener` - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**  
 \* `propertyName` - The name of the property that was listened on.  
 \* `listener` - The PropertyChangeListener to be removed
- 
- *setBeginColumn*  
`public void setBeginColumn( int i )`  
    - **Usage**  
 \* Sets the begin column
- 
- *setBeginLine*  
`public void setBeginLine( int i )`

- **Usage**
  - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

#### 21.2.14 CLASS BitOrAssignExpression

---

This class represents the bit and assign expression nodes of the syntax tree

##### DECLARATION

---

```
public class BitOrAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

##### CONSTRUCTORS

---

- *BitOrAssignExpression*

```
public BitOrAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **lexp** - the LHS expression
      - \* **rexp** - the RHS expression
-

- *BitOrAssignExpression*

```
public BitOrAssignExpression( koala.dynamicjava.tree.Expression lexp,
    koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
    bc, int el, int ec )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression
    - \* **rexp** - the RHS expression
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---

( in 21.2.10, page 1417)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**

- \* Returns the left hand side expression

---

- *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**

- \* Returns the right hand side expression

---

- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

- **Usage**

- \* Sets the left hand side expression

- **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* `java.lang.IllegalArgumentException` - if exp is null

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* `visitor` - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* `listener` - The `PropertyChangeListener` to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
    - **Parameters**
      - \* `propertyName` - The name of the property to listen on.
      - \* `listener` - The `PropertyChangeListener` to be added
- 
- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

    - **Usage**
      - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

- **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*  
**public int getEndLine( )**
  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*  
**public String getFilename( )**
  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*  
**public Set getProperties( )**

- **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string
- 
- *getProperty*

```
public Object getProperty( java.lang.String name )
```

    - **Usage**
      - \* Returns the value of a property
    - **Parameters**
      - \* **name** - the property name
    - **Returns** - null if the property was not previously set
- 
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* **name** - the name of the property
- 
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**
      - \* **propertyName** - The name of the property that was listened on.
      - \* **listener** - The PropertyChangeListener to be removed
- 
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

    - **Usage**
      - \* Sets the begin column
- 
- *setBeginLine*

```
public void setBeginLine( int i )
```

    - **Usage**
      - \* Sets the begin line
- 
- *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
-



- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.15 CLASS BitOrExpression

---

This class represents the bit or expression nodes of the syntax tree

#### DECLARATION

---

```
public class BitOrExpression
extends koala.dynamicjava.tree.BinaryExpression
```

#### CONSTRUCTORS

---

- *BitOrExpression*  
 public **BitOrExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**  
   \* **lexp** - the LHS expression  
   \* **rexp** - the RHS expression  


---
- *BitOrExpression*  
 public **BitOrExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int  
 bc, int el, int ec )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**

- \* Returns the left hand side expression

---

- *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**

- \* Returns the right hand side expression

---

- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression  exp )
```

- **Usage**

- \* Sets the left hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null

---

- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression  exp )
```

- **Usage**

- \* Sets the right hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* **listener** - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

## – Parameters

\* **propertyName** - The name of the property to listen on.  
\* **listener** - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.16 CLASS BlockStatement

---

This class represents the block statement nodes of the syntax tree

#### DECLARATION

---

```
public class BlockStatement
extends koala.dynamicjava.tree.Statement
```

#### FIELDS

---

- public static final String STATEMENTS
  - The creationType property name
- private List statements
  - The list of the statements contained in this block

#### CONSTRUCTORS

---

- *BlockStatement*

```
public BlockStatement( java.util.List  stmts )
```

  - **Usage**
    - \* Creates a new block statement
  - **Parameters**
    - \* **stmts** - the list of the statements contained in this block

---
- *BlockStatement*

```
public BlockStatement( java.util.List  stmts, java.lang.String  fn, int
bl, int  bc, int  el, int  ec )
```

  - **Usage**
    - \* Creates a new block statement
  - **Parameters**
    - \* **stmts** - the list of the statements contained in this block
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getStatements*

```
public List getStatements( )
```

- **Usage**

- \* Returns the statements contained in this block

---

- *setStatements*

```
public void setStatements( java.util.List  l )
```

- **Usage**

- \* Sets the statements contained in this block

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if `l` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`


---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`


---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *addChangeListener*

```
public void addChangeListener( java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* **listener** - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
-



- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
   \* Returns the end column of this node in the end line  


---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
   \* Returns the end line of this node in the source code  


---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
   \* Returns the filename. Can be null.  


---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
   \* Returns the defined properties for this node.  
 – **Returns** - a set of string  


---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns the value of a property  
 – **Parameters**  
   \* name - the property name  
 – **Returns** - null if the property was not previously set  


---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns true if a property is defined for this node  
 – **Parameters**  
   \* name - the name of the property  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener from the listener list.  
   This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be removed  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**

- \* **propertyName** - The name of the property that was listened on.
  - \* **listener** - The PropertyChangeListener to be removed
- - *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
 \* Sets the begin column
- - *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
 \* Sets the begin line
- - *setEndColumn*  
 public void **setEndColumn**( int i )  
 – **Usage**  
 \* Sets the end column
- - *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
 \* Sets the end line
- - *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
 \* Sets the filename
- - *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
 \* Sets the value of a property  
 – **Parameters**  
 \* **name** - the property name  
 \* **value** - the new value to set

### 21.2.17 CLASS BooleanLiteral

---

This class represents the boolean literal nodes of the syntax tree

#### DECLARATION

---

```
public class BooleanLiteral
extends koala.dynamicjava.tree.Literal
```

FIELDS

---

- private static final String TRUE
  - The representations of true
- private static final String FALSE
  - The representations of false

CONSTRUCTORS

---

- *BooleanLiteral*  
 public **BooleanLiteral**( boolean val )
  - **Usage**
    - \* Initializes a literal
  - **Parameters**
    - \* val - the value of the literal

---
- *BooleanLiteral*  
 public **BooleanLiteral**( boolean val, java.lang.String fn, int bl, int bc, int el, int ec )
  - **Usage**
    - \* Initializes a literal
  - **Parameters**
    - \* val - the value of the literal
    - \* fn - the filename
    - \* bl - the begin line
    - \* bc - the begin column
    - \* el - the end line
    - \* ec - the end column

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Literal

---

( in 21.2.67, page 1691)

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* visitor - the visitor to accept

---
- *getRepresentation*  
 public String **getRepresentation**( )
  - **Usage**
    - \* Returns the representation of this object

---

- *getType*  
 public Class **getType**( )  
 – **Usage**  
 \* Returns the type of this expression.  
 NOTE: the 'null' literal has a null type
 

---
- *getValue*  
 public Object **getValue**( )  
 – **Usage**  
 \* Returns the value of this expression
 

---
- *setRepresentation*  
 public void **setRepresentation**( java.lang.String s )  
 – **Usage**  
 \* Sets the representation of this object  
 – **Exceptions**  
 \* java.lang.IllegalArgumentException - if s is null
 

---
- *setType*  
 public void **setType**( java.lang.Class c )  
 – **Usage**  
 \* Sets the type of this object
 

---
- *setValue*  
 public void **setValue**( java.lang.Object o )  
 – **Usage**  
 \* Sets the value of this object  
 – **Exceptions**  
 \* java.lang.IllegalArgumentException - if o is null
 

---

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
 \* Allows a visitor to traverse the tree  
 – **Parameters**  
 \* **visitor** - the visitor to accept
 

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.
    - The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added
- 

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
-

- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.

- **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.18 CLASS BooleanType

---

This class represents the boolean type nodes of the syntax tree

## DECLARATION

---

```
public class BooleanType
extends koala.dynamicjava.tree.PrimitiveType
```

---

## CONSTRUCTORS

- *BooleanType*  
 public **BooleanType**( )  
 — **Usage**  
   \* Initializes the type

---

- *BooleanType*  
 public **BooleanType**( java.lang.String fn, int bl, int bc, int el, int ec )  
 — **Usage**  
   \* Initializes the type  
 — **Parameters**  
   \* **fn** - the filename  
   \* **bl** - the begin line  
   \* **bc** - the begin column  
   \* **el** - the end line  
   \* **ec** - the end column

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimitiveType

---

( in 21.2.89, page 1795)

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 — **Usage**  
   \* Allows a visitor to traverse the tree  
 — **Parameters**  
   \* **visitor** - the visitor to accept

---

- *getValue*  
 public Class **getValue**( )  
 — **Usage**  
   \* Returns the value of this node

---

- *setValue*  
 public void **setValue**( java.lang.Class c )  
 — **Usage**  
   \* Sets the value of this node  
 — **Exceptions**  
   \* java.lang.IllegalArgumentException - if c is null



METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*  
**public int getEndLine( )**
  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*  
**public String getFilename( )**
  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*  
**public Set getProperties( )**
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---

- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.19 CLASS BreakStatement

---

This class represents the break statement nodes of the syntax tree

#### DECLARATION

---

```
public class BreakStatement
extends koala.dynamicjava.tree.Statement
```

#### FIELDS

---

- public static final String LABEL
  - The label property name
- private String label
  - The label

#### CONSTRUCTORS

---

- *BreakStatement*  
 public **BreakStatement**( java.lang.String label )
    - **Usage**
      - \* Creates a new while statement
    - **Parameters**
      - \* **label** - the label
- 
- *BreakStatement*  
 public **BreakStatement**( java.lang.String label, java.lang.String fn, int bl, int bc, int el, int ec )
    - **Usage**
      - \* Creates a new while statement
    - **Parameters**
      - \* **label** - the label
      - \* **fn** - the filename

\* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

\* Allows a visitor to traverse the tree

- **Parameters**

\* **visitor** - the visitor to accept

---

- *getLabel*

```
public String getLabel( )
```

- **Usage**

\* Gets the label

---

- *setLabel*

```
public void setLabel( java.lang.String  s )
```

- **Usage**

\* Sets the label

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Statement

---

( in 21.2.102, page 1857)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

\* Allows a visitor to traverse the tree

- **Parameters**

\* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener )
```

- **Usage**

\* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.

- **Parameters**
    - \* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*

```
public String getFilename( )
```

  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*

```
public Set getProperties( )
```

  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**

- \* Removes a PropertyChangeListener for a specific property.
- **Parameters**
  - \* **propertyName** - The name of the property that was listened on.
  - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.20 CLASS ByteType

---

This class represents the byte type nodes of the syntax tree

#### DECLARATION

---

```
public class ByteType
extends koala.dynamicjava.tree.PrimitiveType
```



CONSTRUCTORS

---

- *ByteType*  
**public ByteType( )**
  - **Usage**
    - \* Initializes the type

---
- *ByteType*  
**public ByteType( java.lang.String fn, int bl, int bc, int el, int ec )**
  - **Usage**
    - \* Initializes the type
  - **Parameters**
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS INHERITED FROM CLASS *koala.dynamicjava.tree.PrimitiveType*

---

( in 21.2.89, page 1795)

- *acceptVisitor*  
**public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

---
- *getValue*  
**public Class getValue( )**
  - **Usage**
    - \* Returns the value of this node

---
- *setValue*  
**public void setValue( java.lang.Class c )**
  - **Usage**
    - \* Sets the value of this node
  - **Exceptions**
    - \* *java.lang.IllegalArgumentException* - if *c* is null

METHODS INHERITED FROM CLASS *koala.dynamicjava.tree.Type*

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.21 CLASS CastExpression

---

This class represents the cast expression nodes of the syntax tree

#### DECLARATION

---

```
public class CastExpression
extends koala.dynamicjava.tree.UnaryExpression
```

#### FIELDS

---

- public static final String TARGET\_TYPE
  - The targetType property name
- private Type targetType
  - The target type

#### CONSTRUCTORS

---

- *CastExpression*

```
public CastExpression( koala.dynamicjava.tree.Type tt,
koala.dynamicjava.tree.Expression exp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **tt** - the target type
    - \* **exp** - the casted expression
- *CastExpression*

```
public CastExpression( koala.dynamicjava.tree.Type tt,
koala.dynamicjava.tree.Expression exp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **tt** - the target type
    - \* **exp** - the casted expression
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getTargetType*

```
public Type getTargetType( )
```

– **Usage**

\* Returns the target type

---

• *setTargetType*

```
public void setTargetType( koala.dynamicjava.tree.Type t )
```

– **Usage**

\* Sets the target type

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if t is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.UnaryExpression`

---

( in 21.2.120, page 1938)

• *getExpression*

```
public Expression getExpression( )
```

– **Usage**

\* Returns the target expression

---

• *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression e )
```

– **Usage**

\* Sets the target expression

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**



- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.22 CLASS *CatchStatement*

---

To store the informations about the catch statements

#### DECLARATION

---

```
public class CatchStatement
extends koala.dynamicjava.tree.Statement
```

#### FIELDS

---

- private FormalParameter exception
  - The caught exception
- private Node block
  - The block

#### CONSTRUCTORS

---

- *CatchStatement*  

```
public CatchStatement( koala.dynamicjava.tree.FormalParameter fp,
koala.dynamicjava.tree.Node blk, java.lang.String fn, int bl, int bc, int
el, int ec )
```

  - **Usage**
    - \* Creates a new catch statement
  - **Parameters**
    - \* **fp** - the caught exception
    - \* **blk** - the block
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

#### METHODS

---

- *acceptVisitor*  

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* **visitor** - the visitor to accept
-

- *getBlock*  
**public Node getBlock( )**  
  - **Usage**  
 \* Returns the block

---
- *getException*  
**public FormalParameter getException( )**  
  - **Usage**  
 \* Returns the caught exception descriptor

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`

---

( in 21.2.102, page 1857)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
**public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**  
  - **Usage**  
 \* Allows a visitor to traverse the tree
  - **Parameters**  
 \* `visitor` - the visitor to accept

---
- *addPropertyChangeListener*  
**public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.
  - **Parameters**  
 \* `listener` - The PropertyChangeListener to be added

---
- *addPropertyChangeListener*  
**public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**  
 \* `propertyName` - The name of the property to listen on.  
 \* `listener` - The PropertyChangeListener to be added

---
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**

- **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *getBeginColumn*  
public int **getBeginColumn**( )
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  
public int **getBeginLine**( )
  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  
public int **getEndColumn**( )
  - **Usage**
    - \* Returns the end column of this node in the end line

---
- *getEndLine*  
public int **getEndLine**( )
  - **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  
public String **getFilename**( )
  - **Usage**
    - \* Returns the filename. Can be null.

---

- 
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
   \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns the value of a property  
 – **Parameters**  
   \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns true if a property is defined for this node  
 – **Parameters**  
   \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener from the listener list.  
   This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* propertyName - The name of the property that was listened on.  
   \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
   \* Sets the begin column

---

  - *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
   \* Sets the begin line

---

  - *setEndColumn*  
 public void **setEndColumn**( int i )

- **Usage**
    - \* Sets the end column
- 
- *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- 
- *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.23 CLASS *CharacterLiteral*

This class represents the character literal nodes of the syntax tree

#### DECLARATION

```
public class CharacterLiteral
extends koala.dynamicjava.tree.Literal
```

#### CONSTRUCTORS

- *CharacterLiteral*

```
public CharacterLiteral( java.lang.String rep )
```

    - **Usage**
      - \* Initializes a literal
    - **Parameters**
      - \* **rep** - the representation of the literal
      - \* **val** - the value of this literal
- 
- *CharacterLiteral*

```
public CharacterLiteral( java.lang.String rep, java.lang.String fn, int
bl, int bc, int el, int ec )
```

    - **Usage**
      - \* Initializes a literal
    - **Parameters**

- \* **rep** - the representation of the literal
- \* **val** - the value of this literal
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *decodeCharacter*

```
private static char decodeCharacter( java.lang.String rep )
```

- **Usage**

- \* Decodes the representation of a Java literal character.  
The input is not checked since this method always called  
on a string produced by the parser.

- **Parameters**

- \* **rep** - the representation of the character

- **Returns** - the character represented by the given string

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Literal

---

( in 21.2.67, page 1691)

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getRepresentation*

```
public String getRepresentation( )
```

- **Usage**

- \* Returns the representation of this object

---

- *getType*

```
public Class getType( )
```

- **Usage**

- \* Returns the type of this expression.  
NOTE: the 'null' literal has a null type

---

- *getValue*

```
public Object getValue( )
```

- **Usage**

- \* Returns the value of this expression

---

- *setRepresentation*

```
public void setRepresentation( java.lang.String s )
```

- **Usage**

- \* Sets the representation of this object
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if `s` is null
- *setType*


---

```
public void setType( java.lang.Class c )
```

  - **Usage**
    - \* Sets the type of this object
- *setValue*


---

```
public void setValue( java.lang.Object o )
```

  - **Usage**
    - \* Sets the value of this object
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if `o` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*


---

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added
- *addPropertyChangeListener*


---

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**



- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

– **Parameters**

- \* **propertyName** - The name of the property to listen on.
- \* **listener** - The PropertyChangeListener to be added

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
   \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
   \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
   \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns the value of a property  
 – **Parameters**  
   \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns true if a property is defined for this node  
 – **Parameters**  
   \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener from the listener list.  
   This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* propertyName - The name of the property that was listened on.  
   \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )

- **Usage**
  - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
  - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
  - \* Sets the end column

---
- *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
  - \* Sets the end line

---
- *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
  - \* Sets the filename

---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
  - \* Sets the value of a property
  - **Parameters**
  - \* **name** - the property name
  - \* **value** - the new value to set

### 21.2.24 CLASS CharType

---

This class represents the char type nodes of the syntax tree

#### DECLARATION

---

```
public class CharType
extends koala.dynamicjava.tree.PrimitiveType
```

#### CONSTRUCTORS

---

- *CharType*  
 public **CharType**( )
  - **Usage**
  - \* Initializes the type

---

- *CharType*

```
public CharType( java.lang.String fn, int bl, int bc, int el, int ec )
```

- **Usage**

- \* Initializes the type

- **Parameters**

- \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimitiveType

---

( in 21.2.89, page 1795)

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getValue*

```
public Class getValue( )
```

- **Usage**

- \* Returns the value of this node

---

- *setValue*

```
public void setValue( java.lang.Class c )
```

- **Usage**

- \* Sets the value of this node

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if `c` is null

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Type

---

( in 21.2.117, page 1924)

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

\* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.
  - The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
  - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
  - \* **oldValue** - The old value of the property.
  - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue,
int newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
  - \* **oldValue** - The old value of the property.
  - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName,
java.lang.Object oldValue, java.lang.Object newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
  - \* **oldValue** - The old value of the property.
  - \* **newValue** - The new value of the property.

- 
- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**

- \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
- **Parameters**
  - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.25 CLASS ClassAllocation

---

This class represents the anonymous class allocation nodes of the syntax tree

DECLARATION

---

```
public class ClassAllocation
extends koala.dynamicjava.tree.Allocation
implements ExpressionStatement
```

FIELDS

---

- public static final String ARGUMENTS
  - The arguments property name
- public static final String MEMBERS
  - The members property name
- private List arguments
  - The arguments to pass to the constructor
- private List members
  - The members of the anonymous class

CONSTRUCTORS

---

- *ClassAllocation*

```
public ClassAllocation( koala.dynamicjava.tree.Type  tp, java.util.List
args, java.util.List  memb )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **tp** - the type prefix
    - \* **args** - the arguments of the constructor. Can be null.
    - \* **memb** - the members of the class

---
- *ClassAllocation*

```
public ClassAllocation( koala.dynamicjava.tree.Type  tp, java.util.List
args, java.util.List  memb, java.lang.String  fn, int  bl, int  bc, int
el, int  ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **tp** - the type prefix
    - \* **args** - the arguments of the constructor. null if no arguments.
    - \* **memb** - the members of the class
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column



## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getArguments*

```
public List getArguments( )
```

- **Usage**

- \* Returns the constructor arguments

- **Returns** - null if there is no arguments

---

- *getMembers*

```
public List getMembers( )
```

- **Usage**

- \* Returns the members of the anonymous class

---

- *setArguments*

```
public void setArguments( java.util.List  l )
```

- **Usage**

- \* Sets the constructor arguments

---

- *setMembers*

```
public void setMembers( java.util.List  l )
```

- **Usage**

- \* Sets the members of the anonymous class

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if t is null

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Allocation`


---

( in 21.2.3, page 1387)

- *getCreationType*

```
public Type getCreationType( )
```

- **Usage**

- \* Returns the creation type

---

- *setCreationType*

```
public void setCreationType( koala.dynamicjava.tree.Type  t )
```

- **Usage**

- \* Sets the creation type

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if t is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* **listener** - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

## – Parameters

\* **propertyName** - The name of the property to listen on.  
\* **listener** - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
**public void setEndColumn( int i )**
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
**public void setEndLine( int i )**
  - **Usage**

- \* Sets the end line
- - *setFilename*  
`public void setFilename( java.lang.String s )`
    - **Usage**
      - \* Sets the filename

---
- *setProperty*  
`public void setProperty( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.26 CLASS ClassDeclaration

---

This class represents a class declaration

#### DECLARATION

---

```
public class ClassDeclaration
extends koala.dynamicjava.tree.TypeDeclaration
```

#### FIELDS

---

- public static final String SUPERCLASS
  - The superclass property name
- private String superclass
  - The superclass of this class

#### CONSTRUCTORS

---

- *ClassDeclaration*  
`public ClassDeclaration( int flags, java.lang.String name, java.util.List ext, java.util.List impl, java.util.List body )`
  - **Usage**
    - \* Creates a new class declaration
  - **Parameters**
    - \* **flags** - the access flags
    - \* **name** - the name of the class to declare
    - \* **ext** - the tokens that compose the name of the parent class.  
 The list can be null. The superclass property is then set to "java.lang.Object".

- \* **impl** - the list of implemented interfaces (a list of list of Token). Can be null.
- \* **body** - the list of members declarations

---

- *ClassDeclaration*

```
public ClassDeclaration( int  flags, java.lang.String  name, java.util.List
ext, java.util.List  impl, java.util.List  body, java.lang.String  fn, int
bl, int  bc, int  el, int  ec )
```

- **Usage**

- \* Creates a new class declaration

- **Parameters**

- \* **flags** - the access flags
- \* **name** - the name of the class to declare
- \* **ext** - the tokens that compose the name of the parent class.  
The list can be null. The superclass property is then set to "java.lang.Object".
- \* **impl** - the list of implemented interfaces (a list of list of Token). Can be null.
- \* **body** - the list of members declarations
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getSuperclass*

```
public String getSuperclass( )
```

- **Usage**

- \* Returns the name of the superclass of this class

---

- *setSuperclass*

```
public void setSuperclass( java.lang.String  s )
```

- **Usage**

- \* Sets the superclass name

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if s is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.TypeDeclaration`

---

( in 21.2.118, page 1928)

- *getAccessFlags*  
`public int getAccessFlags( )`  
  - **Usage**  
\* Returns the access flags for this class

---

- *getInterfaces*  
`public List getInterfaces( )`  
  - **Usage**  
\* Returns a list that contains the names (String) of the implemented interfaces.  
Can be null.

---

- *getMembers*  
`public List getMembers( )`  
  - **Usage**  
\* Returns the list of the declared members

---

- *getName*  
`public String getName( )`  
  - **Usage**  
\* Returns the name of this class

---

- *setAccessFlags*  
`public void setAccessFlags( int f )`  
  - **Usage**  
\* Sets the access flags for this constructor

---

- *setInterfaces*  
`public void setInterfaces( java.util.List l )`  
  - **Usage**  
\* Sets the interfaces (a list of strings)

---

- *setMembers*  
`public void setMembers( java.util.List l )`  
  - **Usage**  
\* Sets the members
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if l is null

---

- *setName*  
`public void setName( java.lang.String s )`  
  - **Usage**  
\* Sets the type's name
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if s is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.



- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.27 CLASS *ClassInitializer*

---

This class represents the class initializer statement nodes of the syntax tree

#### DECLARATION

---

```
public class ClassInitializer
extends koala.dynamicjava.tree.Initializer
```

#### CONSTRUCTORS

---

- *ClassInitializer*  

```
public ClassInitializer( koala.dynamicjava.tree.BlockStatement  block )
```

  - **Usage**
    - \* Creates a new initializer statement
  - **Parameters**
    - \* **block** - the block

---
- *ClassInitializer*  

```
public ClassInitializer( koala.dynamicjava.tree.BlockStatement  block,
java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

  - **Usage**
    - \* Creates a new initializer statement
  - **Parameters**
    - \* **block** - the block
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

#### METHODS

---

- *acceptVisitor*  

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Initializer`

---

( in 21.2.56, page 1637)

- *getBlock*  
`public BlockStatement getBlock( )`  
    - **Usage**  
 \* Gets the block statement
- 
- *setBlock*  
`public void setBlock( koala.dynamicjava.tree.BlockStatement bs )`  
    - **Usage**  
 \* Sets the block statement
    - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if bs is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
    - **Usage**  
 \* Allows a visitor to traverse the tree
    - **Parameters**  
 \* `visitor` - the visitor to accept
- 
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.
    - **Parameters**  
 \* `listener` - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
    - **Parameters**  
 \* `propertyName` - The name of the property to listen on.  
 \* `listener` - The PropertyChangeListener to be added
- 
- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )`  
    - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.
-

- *getProperties*  
**public Set getProperties( )**
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
**public void setEndColumn( int i )**

- **Usage**
  - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.28 CLASS ComplementExpression

---

This class represents the complement expression nodes of the syntax tree

#### DECLARATION

---

```
public class ComplementExpression
extends koala.dynamicjava.tree.UnaryExpression
```

#### CONSTRUCTORS

---

- *ComplementExpression*

```
public ComplementExpression( koala.dynamicjava.tree.Expression exp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the target expression
- *ComplementExpression*

```
public ComplementExpression( koala.dynamicjava.tree.Expression exp,
java.lang.String fn, int bl, int bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the target expression

- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor **visitor** )  
  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.UnaryExpression

---

( in 21.2.120, page 1938)

- *getExpression*  
 public Expression **getExpression**( )  
  
 – **Usage**  
   \* Returns the target expression

---

- *setExpression*  
 public void **setExpression**( koala.dynamicjava.tree.Expression **e** )  
  
 – **Usage**  
   \* Sets the target expression  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if e is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor **visitor** )  
  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* **visitor** - the visitor to accept
-



- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.
    - The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added
- 

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
-

- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.

- **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.29 CLASS ConditionalExpression

---

This class represents the binary expression nodes of the syntax tree

DECLARATION

---

```
public class ConditionalExpression
extends koala.dynamicjava.tree.Expression
```

FIELDS

---

- public static final String CONDITION\_EXPRESSION
  - The conditionExpression property name
- public static final String IF\_TRUE\_EXPRESSION
  - The ifTrueExpression property name
- public static final String IF\_FALSE\_EXPRESSION
  - The ifFalseExpression property name
- private Expression conditionExpression
  - The condition expression
- private Expression ifTrueExpression
  - The if true expression
- private Expression ifFalseExpression
  - The if false expression

CONSTRUCTORS

---

- *ConditionalExpression*

```
public ConditionalExpression( koala.dynamicjava.tree.Expression cexp,
koala.dynamicjava.tree.Expression texp, koala.dynamicjava.tree.Expression
fexp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **cexp** - the condition expression
    - \* **texp** - the if true expression
    - \* **fexp** - the if false expression

---
- *ConditionalExpression*

```
public ConditionalExpression( koala.dynamicjava.tree.Expression cexp,
koala.dynamicjava.tree.Expression texp, koala.dynamicjava.tree.Expression
fexp, java.lang.String fn, int bl, int bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**

- \* **cexp** - the condition expression
- \* **texp** - the if true expression
- \* **fexp** - the if false expression
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getConditionExpression*

```
public Expression getConditionExpression( )
```

- **Usage**

- \* Returns the condition expression

---

- *getIfFalseExpression*

```
public Expression getIfFalseExpression( )
```

- **Usage**

- \* Returns the if false expression

---

- *getIfTrueExpression*

```
public Expression getIfTrueExpression( )
```

- **Usage**

- \* Returns the if true expression

---

- *setConditionExpression*

```
public void setConditionExpression( koala.dynamicjava.tree.Expression  e )
```

- **Usage**

- \* Sets the condition expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if e is null

---

- *setIfFalseExpression*

```
public void setIfFalseExpression( koala.dynamicjava.tree.Expression  e )
```

- **Usage**

- \* Sets the if false expression

- **Exceptions**

---

\* java.lang.IllegalArgumentException - if e is null

---

- *setIfTrueExpression*

public void **setIfTrueExpression**( koala.dynamicjava.tree.Expression e )

- **Usage**

- \* Sets the if true expression

- **Exceptions**

- \* java.lang.IllegalArgumentException - if e is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* visitor - the visitor to accept

---

- *addPropertyChangeListener*

public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* listener - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* propertyName - The name of the property to listen on.
  - \* listener - The PropertyChangeListener to be added

---

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String propertyName, boolean oldValue, boolean newValue )

- **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

- **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---
- *getEndLine*  
**public int getEndLine( )**
  - **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  
**public String getFilename( )**
  - **Usage**
    - \* Returns the filename. Can be null.

---
- *getProperties*  
**public Set getProperties( )**

- **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* name - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* name - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* listener - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* propertyName - The name of the property that was listened on.
    - \* listener - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---



- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.30 CLASS ConstructorDeclaration

---

This class represents constructor declarations in an AST

#### DECLARATION

---

```
public class ConstructorDeclaration
extends koala.dynamicjava.tree.Node
```

#### FIELDS

---

- public static final String ACCESS\_FLAGS  
 – The accessFlags property name
- public static final String NAME  
 – The name property name
- public static final String PARAMETERS  
 – The parameters property name
- public static final String EXCEPTIONS  
 – The exceptions property name
- public static final String STATEMENTS  
 – The statements property name
- private int accessFlags  
 – The access flags

- private String name
  - The name of this constructor
- private List parameters
  - The parameters
- private List exceptions
  - The exceptions
- private ConstructorInvocation constructorInvocation
  - The explicit constructor invocation
- private List statements
  - The statements

## CONSTRUCTORS

---

- *ConstructorDeclaration*

```
public ConstructorDeclaration( int flags, java.lang.String name,
    java.util.List params, java.util.List excepts,
    koala.dynamicjava.tree.ConstructorInvocation eci, java.util.List stmts )
```

    - **Usage**
      - \* Creates a new method declaration
    - **Parameters**
      - \* **flags** - the access flags
      - \* **name** - the name of this constructor
      - \* **params** - the parameters list
      - \* **excepts** - the exception list (a list of list of token)
      - \* **eci** - the explicit constructor invocation
      - \* **stmts** - the statements
- 
- *ConstructorDeclaration*

```
public ConstructorDeclaration( int flags, java.lang.String name,
    java.util.List params, java.util.List excepts,
    koala.dynamicjava.tree.ConstructorInvocation eci, java.util.List stmts,
    java.lang.String fn, int bl, int bc, int el, int ec )
```

    - **Usage**
      - \* Creates a new method declaration
    - **Parameters**
      - \* **flags** - the access flags
      - \* **name** - the name of this constructor
      - \* **params** - the parameters list
      - \* **excepts** - the exception list (a list of list of token)
      - \* **eci** - the explicit constructor invocation
      - \* **stmts** - the statements
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getAccessFlags*

```
public int getAccessFlags( )
```

– **Usage**

\* Returns the access flags for this constructor

---

• *getConstructorInvocation*

```
public ConstructorInvocation getConstructorInvocation( )
```

– **Usage**

\* The explicit constructor invocation if one or null

---

• *getExceptions*

```
public List getExceptions( )
```

– **Usage**

\* Returns the list of the exception thrown by this method

– **Returns** - a list of string• *getName*

```
public String getName( )
```

– **Usage**

\* Returns the name of this constructor

---

• *getParameters*

```
public List getParameters( )
```

– **Usage**

\* Returns the parameters list

---

• *getStatements*

```
public List getStatements( )
```

– **Usage**

\* Returns the statements

---

• *setAccessFlags*

```
public void setAccessFlags( int  f )
```

– **Usage**

\* Sets the access flags for this constructor

---

- *setConstructorInvocation*

```
public void setConstructorInvocation(
    koala.dynamicjava.tree.ConstructorInvocation ci )
```

- **Usage**

\* Sets the constructor invocation

---

- *setExceptions*

```
public void setExceptions( java.util.List l )
```

- **Usage**

\* Sets the exceptions thrown by this method

- **Parameters**

\* l - a list of string

- **Exceptions**

\* java.lang.IllegalArgumentException - if l is null

---

- *setName*

```
public void setName( java.lang.String s )
```

- **Usage**

\* Sets the constructor's name

- **Exceptions**

\* java.lang.IllegalArgumentException - if s is null

---

- *setParameters*

```
public void setParameters( java.util.List l )
```

- **Usage**

\* Sets the parameters

---

- *setStatements*

```
public void setStatements( java.util.List l )
```

- **Usage**

\* Sets the statements

- **Exceptions**

\* java.lang.IllegalArgumentException - if l is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor
)
```

- **Usage**

\* Allows a visitor to traverse the tree

- **Parameters**

\* visitor - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

- Usage

- \* Adds a PropertyChangeListener to the listener list.
  - The listener is registered for all properties.

- Parameters

- \* listener - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

- Usage

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- Parameters

- \* propertyName - The name of the property to listen on.
  - \* listener - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
  - \* oldValue - The old value of the property.
  - \* newValue - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue,
int newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
  - \* oldValue - The old value of the property.
  - \* newValue - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName,
java.lang.Object oldValue, java.lang.Object newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
  - \* oldValue - The old value of the property.
  - \* newValue - The new value of the property.

- 
- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**

- \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
- **Parameters**
  - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*  

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*  

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.31 CLASS ConstructorInvocation

---

This class represents the constructor call nodes of the syntax tree

DECLARATION

---

```
public class ConstructorInvocation
extends koala.dynamicjava.tree.PrimaryExpression
implements ExpressionContainer
```

FIELDS

---

- public static final String ARGUMENTS
  - The arguments property name
- public static final String SUPER
  - The super property name
- private Expression expression
  - The prefix expression
- private List arguments
  - The arguments
- private boolean superCall
  - Whether this invocation is 'super' or 'this'

CONSTRUCTORS

---

- *ConstructorInvocation*

```
public ConstructorInvocation( koala.dynamicjava.tree.Expression exp,
                             java.util.List args, boolean sup )
```

  - **Usage**
    - \* Creates a new node
  - **Parameters**
    - \* **exp** - the prefix expression
    - \* **args** - the arguments. null if there are no argument.
    - \* **sup** - whether this invocation is 'super' or 'this'
- *ConstructorInvocation*

```
public ConstructorInvocation( koala.dynamicjava.tree.Expression exp,
                             java.util.List args, boolean sup, java.lang.String fn, int bl, int bc,
                             int el, int ec )
```

  - **Usage**
    - \* Creates a new node
  - **Parameters**
    - \* **exp** - the prefix expression
    - \* **args** - the arguments. null if there are no argument.
    - \* **sup** - whether this invocation is 'super' or 'this'



\* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getArguments*

```
public List getArguments( )
```

- **Usage**

- \* Returns the arguments

---

- *getExpression*

```
public Expression getExpression( )
```

- **Usage**

- \* Returns the prefix expression if one, or null otherwise

---

- *isSuper*

```
public boolean isSuper( )
```

- **Usage**

- \* Returns true if this invocation is a 'super' or a 'this' invocation

---

- *setArguments*

```
public void setArguments( java.util.List  l )
```

- **Usage**

- \* Sets the arguments

---

- *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression  e )
```

- **Usage**

- \* Sets the prefix expression

---

- *setSuper*

```
public void setSuper( boolean  b )
```

- **Usage**

- \* Sets the super property

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
**public void setEndColumn( int i )**
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
**public void setEndLine( int i )**
  - **Usage**

- \* Sets the end line
- - *setFilename*  
`public void setFilename( java.lang.String s )`
    - **Usage**
      - \* Sets the filename

---
- *setProperty*  
`public void setProperty( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.32 CLASS ContinueStatement

---

This class represents the continue statement nodes of the syntax tree

#### DECLARATION

---

```
public class ContinueStatement
extends koala.dynamicjava.tree.Statement
```

#### FIELDS

---

- public static final String LABEL
  - The label property name
- private String label
  - The label

#### CONSTRUCTORS

---

- *ContinueStatement*  
`public ContinueStatement( java.lang.String label )`
  - **Usage**
    - \* Creates a new while statement
  - **Parameters**
    - \* **label** - the label
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

---

- *ContinueStatement*

```
public ContinueStatement( java.lang.String label, java.lang.String fn, int
bl, int bc, int el, int ec )
```

- **Usage**

- \* Creates a new while statement

- **Parameters**

- \* **label** - the label
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getLabel*

```
public String getLabel( )
```

- **Usage**

- \* Gets the label

---

- *setLabel*

```
public void setLabel( java.lang.String s )
```

- **Usage**

- \* Sets the label

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Statement

---

( in 21.2.102, page 1857)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor
)
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**
    - \* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener  
will be invoked only when a call on firePropertyChange names that  
specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue,
int newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName,
java.lang.Object oldValue, java.lang.Object newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.

\* **newValue** - The new value of the property.

---

- *getBeginColumn*

public int **getBeginColumn**( )

– **Usage**

\* Returns the begin column of this node in the begin line

---

- *getBeginLine*

public int **getBeginLine**( )

– **Usage**

\* Returns the begin line of this node in the source code

---

- *getEndColumn*

public int **getEndColumn**( )

– **Usage**

\* Returns the end column of this node in the end line

---

- *getEndLine*

public int **getEndLine**( )

– **Usage**

\* Returns the end line of this node in the source code

---

- *getFilename*

public String **getFilename**( )

– **Usage**

\* Returns the filename. Can be null.

---

- *getProperties*

public Set **getProperties**( )

– **Usage**

\* Returns the defined properties for this node.

– **Returns** - a set of string

---

- *getProperty*

public Object **getProperty**( java.lang.String **name** )

– **Usage**

\* Returns the value of a property

– **Parameters**

\* **name** - the property name

– **Returns** - null if the property was not previously set

---

- *hasProperty*

public boolean **hasProperty**( java.lang.String **name** )

– **Usage**

\* Returns true if a property is defined for this node

– **Parameters**

\* **name** - the name of the property

---

- *removePropertyChangeListener*

public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )

– **Usage**



- \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*  

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*  

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.33 CLASS DivideAssignExpression

---

This class represents the divide assign expression nodes of the syntax tree

DECLARATION

---

```
public class DivideAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

CONSTRUCTORS

---

- *DivideAssignExpression*

```
public DivideAssignExpression( koala.dynamicjava.tree.Expression  lexp,
koala.dynamicjava.tree.Expression  rexp )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression
  - \* **rexp** - the RHS expression

---

- *DivideAssignExpression*

```
public DivideAssignExpression( koala.dynamicjava.tree.Expression  lexp,
koala.dynamicjava.tree.Expression  rexp, java.lang.String  fn, int  bl, int
bc, int  el, int  ec )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression
  - \* **rexp** - the RHS expression
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.BinaryExpression`

---

( in 21.2.11, page 1421)

- *getLeftExpression*  
`public Expression getLeftExpression( )`  
  - **Usage**  
\* Returns the left hand side expression
- *getRightExpression*  
`public Expression getRightExpression( )`  
  - **Usage**  
\* Returns the right hand side expression
- *setLeftExpression*  
`public void setLeftExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
\* Sets the left hand side expression
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if exp is null
- *setRightExpression*  
`public void setRightExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
\* Sets the right hand side expression
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
  - **Usage**  
\* Allows a visitor to traverse the tree
  - **Parameters**  
\* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`  
  - **Usage**  
\* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**

\* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- • *getEndColumn*  
 public int **getEndColumn**( )
  - **Usage**
    - \* Returns the end column of this node in the end line
- • *getEndLine*  
 public int **getEndLine**( )
  - **Usage**
    - \* Returns the end line of this node in the source code
- • *getFilename*  
 public String **getFilename**( )
  - **Usage**
    - \* Returns the filename. Can be null.
- • *getProperties*  
 public Set **getProperties**( )
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string
- • *getProperty*  
 public Object **getProperty**( java.lang.String name )
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* name - the property name
  - **Returns** - null if the property was not previously set
- • *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* name - the name of the property
- • *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* listener - The PropertyChangeListener to be removed
- • *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.

- **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.34 CLASS DivideExpression

---

This class represents the divide expression nodes of the syntax tree

#### DECLARATION

---

```
public class DivideExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *DivideExpression*

```
public DivideExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression

---

• *DivideExpression*

```
public DivideExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

## – Usage

\* Returns the left hand side expression

---

• *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.



---

\* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

**public int getBeginColumn( )**

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

**public int getBeginLine( )**

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

**public int getEndColumn( )**

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

**public int getEndLine( )**

- **Usage**

- \* Returns the end line of this node in the source code

- 
- *getFilename*  
`public String getFilename( )`  
    - **Usage**  
 \* Returns the filename. Can be null.
- 
- *getProperties*  
`public Set getProperties( )`  
    - **Usage**  
 \* Returns the defined properties for this node.
    - **Returns** - a set of string
- 
- *getProperty*  
`public Object getProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns the value of a property
    - **Parameters**  
 \* `name` - the property name
    - **Returns** - null if the property was not previously set
- 
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns true if a property is defined for this node
    - **Parameters**  
 \* `name` - the name of the property
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**  
 \* `listener` - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**  
 \* `propertyName` - The name of the property that was listened on.  
 \* `listener` - The PropertyChangeListener to be removed
- 
- *setBeginColumn*  
`public void setBeginColumn( int i )`  
    - **Usage**  
 \* Sets the begin column
- 
- *setBeginLine*  
`public void setBeginLine( int i )`

- **Usage**
    - \* Sets the begin line
- - *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
- - *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- - *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- - *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.35 CLASS DoStatement

---

This class represents the do statement nodes of the syntax tree

#### DECLARATION

---

```
public class DoStatement
extends koala.dynamicjava.tree.Statement
implements ContinueTarget
```

#### FIELDS

---

- public static final String CONDITION
  - The condition property name
- public static final String BODY
  - The body property name
- private Expression condition
  - The condition to evaluate at each loop
- private Node body

- The body of this statement
- private List labels
  - The labels

## CONSTRUCTORS

---

- *DoStatement*

```
public DoStatement( koala.dynamicjava.tree.Expression cond,
koala.dynamicjava.tree.Node body )
```

- Usage

- \* Creates a new do statement

- Parameters

- \* **cond** - the condition to evaluate at each loop
  - \* **body** - the body

---

- *DoStatement*

```
public DoStatement( koala.dynamicjava.tree.Expression cond,
koala.dynamicjava.tree.Node body, java.lang.String fn, int bl, int bc,
int el, int ec )
```

- Usage

- \* Creates a new do statement

- Parameters

- \* **cond** - the condition to evaluate at each loop
  - \* **body** - the body
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- Usage

- \* Allows a visitor to traverse the tree

- Parameters

- \* **visitor** - the visitor to accept

---

- *addLabel*

```
public void addLabel( java.lang.String label )
```

- Usage

- \* Adds a label to this statement

- **Parameters**
    - \* `label` - the label to add
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if `label` is null

---
- *getBody*

```
public Node getBody( )
```

  - **Usage**
    - \* Returns the body of this statement

---
- *getCondition*

```
public Expression getCondition( )
```

  - **Usage**
    - \* Gets the condition to evaluate at each loop

---
- *hasLabel*

```
public boolean hasLabel( java.lang.String label )
```

  - **Usage**
    - \* Test whether this statement has the given label
  - **Returns** - true if this statement has the given label

---
- *setBody*

```
public void setBody( koala.dynamicjava.tree.Node node )
```

  - **Usage**
    - \* Sets the body of this statement
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if `node` is null

---
- *setCondition*

```
public void setCondition( koala.dynamicjava.tree.Expression e )
```

  - **Usage**
    - \* Sets the condition to evaluate
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if `e` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`

---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set



### 21.2.36 CLASS DoubleLiteral

---

This class represents the double literal nodes of the syntax tree

#### DECLARATION

---

```
public class DoubleLiteral
extends koala.dynamicjava.tree.Literal
```

#### CONSTRUCTORS

---

- *DoubleLiteral*  
**public DoubleLiteral( java.lang.String rep )**
  - **Usage**
    - \* Initializes a literal
  - **Parameters**
    - \* **rep** - the representation of the literal

---
- *DoubleLiteral*  
**public DoubleLiteral( java.lang.String rep, java.lang.String fn, int bl, int bc, int el, int ec )**
  - **Usage**
    - \* Initializes a literal
  - **Parameters**
    - \* **rep** - the representation of the literal
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Literal

---

( in 21.2.67, page 1691)

- *acceptVisitor*  
**public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

---
- *getRepresentation*  
**public String getRepresentation( )**
  - **Usage**
    - \* Returns the representation of this object

- 
- *getType*  
 public Class **getType**( )  
 – **Usage**  
 \* Returns the type of this expression.  
 NOTE: the 'null' literal has a null type

---

  - *getValue*  
 public Object **getValue**( )  
 – **Usage**  
 \* Returns the value of this expression

---

  - *setRepresentation*  
 public void **setRepresentation**( java.lang.String s )  
 – **Usage**  
 \* Sets the representation of this object  
 – **Exceptions**  
 \* java.lang.IllegalArgumentException - if s is null

---

  - *setType*  
 public void **setType**( java.lang.Class c )  
 – **Usage**  
 \* Sets the type of this object

---

  - *setValue*  
 public void **setValue**( java.lang.Object o )  
 – **Usage**  
 \* Sets the value of this object  
 – **Exceptions**  
 \* java.lang.IllegalArgumentException - if o is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
 \* Allows a visitor to traverse the tree  
 – **Parameters**  
 \* visitor - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

- Usage

- \* Adds a PropertyChangeListener to the listener list.
    - The listener is registered for all properties.

- Parameters

- \* listener - The PropertyChangeListener to be added
- 

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

- Usage

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- Parameters

- \* propertyName - The name of the property to listen on.
      - \* listener - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
      - \* oldValue - The old value of the property.
      - \* newValue - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue,
int newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
      - \* oldValue - The old value of the property.
      - \* newValue - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName,
java.lang.Object oldValue, java.lang.Object newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
      - \* oldValue - The old value of the property.
      - \* newValue - The new value of the property.
-

- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line  


---
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code  


---
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line  


---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code  


---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.  


---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string  


---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set  


---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.

- **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.37 CLASS DoubleType

---

This class represents the double type nodes of the syntax tree

DECLARATION

---

```
public class DoubleType
extends koala.dynamicjava.tree.PrimitiveType
```

CONSTRUCTORS

---

- *DoubleType*  
 public **DoubleType**( )  
 – **Usage**  
   \* Initializes the type

---

- *DoubleType*  
 public **DoubleType**( java.lang.String fn, int bl, int bc, int el, int ec )  
 – **Usage**  
   \* Initializes the type  
 – **Parameters**  
   \* **fn** - the filename  
   \* **bl** - the begin line  
   \* **bc** - the begin column  
   \* **el** - the end line  
   \* **ec** - the end column

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimitiveType

---

( in 21.2.89, page 1795)

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* **visitor** - the visitor to accept

---

- *getValue*  
 public Class **getValue**( )  
 – **Usage**  
   \* Returns the value of this node

---

- *setValue*  
 public void **setValue**( java.lang.Class c )  
 – **Usage**  
   \* Sets the value of this node  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if c is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property



- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*  
`public void setProperty( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.38 CLASS EmptyStatement

---

This class represents the empty statement nodes of the syntax tree

#### DECLARATION

---

```
public class EmptyStatement
extends koala.dynamicjava.tree.Statement
```

#### CONSTRUCTORS

---

- *EmptyStatement*  
`public EmptyStatement( )`
    - **Usage**
      - \* Creates a new empty statement
- 
- *EmptyStatement*  
`public EmptyStatement( java.lang.String fn, int line, int col )`
    - **Usage**
      - \* Creates a new empty statement
    - **Parameters**
      - \* **line** - the line in the input where the statement occurs
      - \* **fn** - the filename
      - \* **col** - the column

#### METHODS

---

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`

---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

\* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

- **Usage**

- \* Sets the value of a property

- **Parameters**

- \* **name** - the property name

- \* **value** - the new value to set

### 21.2.39 CLASS EqualExpression

---

This class represents the equal expression nodes of the syntax tree

#### DECLARATION

---

```
public class EqualExpression
extends koala.dynamicjava.tree.BinaryExpression
```

#### CONSTRUCTORS

---

- *EqualExpression*

```
public EqualExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression

- \* **rexp** - the RHS expression

---

- *EqualExpression*

```
public EqualExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression

- \* **rexp** - the RHS expression

- \* **fn** - the filename

- \* **bl** - the begin line

- \* **bc** - the begin column

- \* **el** - the end line

- \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

– **Usage**

\* Returns the left hand side expression

• *getRightExpression*

```
public Expression getRightExpression( )
```

– **Usage**

\* Returns the right hand side expression

• *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression  exp )
```

– **Usage**

\* Sets the left hand side expression

– **Exceptions**

\* java.lang.IllegalArgumentException - if exp is null

• *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression  exp )
```

– **Usage**

\* Sets the right hand side expression

– **Exceptions**

\* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.



- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.40 CLASS ExclusiveOrAssignExpression

---

This class represents the exclusive and assign expression nodes of the syntax tree

### DECLARATION

---

```
public class ExclusiveOrAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

### CONSTRUCTORS

---

- *ExclusiveOrAssignExpression*  
 public **ExclusiveOrAssignExpression**( koala.dynamicjava.tree.Expression  
   **lexp**, koala.dynamicjava.tree.Expression **rexp** )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**  
   \* **lexp** - the LHS expression  
   \* **rexp** - the RHS expression

---

- *ExclusiveOrAssignExpression*  
 public **ExclusiveOrAssignExpression**( koala.dynamicjava.tree.Expression  
   **lexp**, koala.dynamicjava.tree.Expression **rexp**, java.lang.String **fn**, int  
   **bl**, int **bc**, int **el**, int **ec** )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**  
   \* **lexp** - the LHS expression  
   \* **rexp** - the RHS expression  
   \* **fn** - the filename  
   \* **bl** - the begin line  
   \* **bc** - the begin column  
   \* **el** - the end line  
   \* **ec** - the end column

### METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor **visitor** )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* **visitor** - the visitor to accept

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.AssignExpression`


---

( in 21.2.10, page 1417)

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.BinaryExpression`


---

( in 21.2.11, page 1421)

- *getLeftExpression*  
`public Expression getLeftExpression( )`  
  - **Usage**  
\* Returns the left hand side expression
- *getRightExpression*  
`public Expression getRightExpression( )`  
  - **Usage**  
\* Returns the right hand side expression
- *setLeftExpression*  
`public void setLeftExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
\* Sets the left hand side expression
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if exp is null
- *setRightExpression*  
`public void setRightExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
\* Sets the right hand side expression
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if exp is null

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`


---

( in 21.2.42, page 1571)

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`


---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
  - **Usage**  
\* Allows a visitor to traverse the tree
  - **Parameters**  
\* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`

- **Usage**
  - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
- **Parameters**
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

- 
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
 \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
 \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
 \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
 \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
 \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
 \* Sets the value of a property  
 – **Parameters**  
 \* **name** - the property name  
 \* **value** - the new value to set

### 21.2.41 CLASS ExclusiveOrExpression

---

This class represents the bit or expression nodes of the syntax tree

#### DECLARATION

---

```
public class ExclusiveOrExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *ExclusiveOrExpression*

```
public ExclusiveOrExpression( koala.dynamicjava.tree.Expression lexp,
                             koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression

---

• *ExclusiveOrExpression*

```
public ExclusiveOrExpression( koala.dynamicjava.tree.Expression lexp,
                             koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
                             bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

## – Usage

\* Returns the left hand side expression

---

• *getRightExpression*

```
public Expression getRightExpression( )
```



- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.

---

\* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

**public int getBeginColumn( )**

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

**public int getBeginLine( )**

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

**public int getEndColumn( )**

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

**public int getEndLine( )**

- **Usage**

- \* Returns the end line of this node in the source code

- 
- *getFilename*  
`public String getFilename( )`  
    - **Usage**  
 \* Returns the filename. Can be null.
- 
- *getProperties*  
`public Set getProperties( )`  
    - **Usage**  
 \* Returns the defined properties for this node.
    - **Returns** - a set of string
- 
- *getProperty*  
`public Object getProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns the value of a property
    - **Parameters**  
 \* **name** - the property name
    - **Returns** - null if the property was not previously set
- 
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns true if a property is defined for this node
    - **Parameters**  
 \* **name** - the name of the property
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**  
 \* **listener** - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed
- 
- *setBeginColumn*  
`public void setBeginColumn( int i )`  
    - **Usage**  
 \* Sets the begin column
- 
- *setBeginLine*  
`public void setBeginLine( int i )`

- **Usage**
    - \* Sets the begin line
- - *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
- - *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- - *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- - *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

## 21.2.42 CLASS Expression

---

This class represents the expression nodes of the syntax tree

### DECLARATION

---

```
public abstract class Expression
extends koala.dynamicjava.tree.Node
```

### CONSTRUCTORS

---

- *Expression*

```
protected Expression( java.lang.String fn, int bl, int bc, int el, int ec )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.43 CLASS FieldAccess

---

This class represents the field access nodes of the syntax tree

#### DECLARATION

---

```
public abstract class FieldAccess
extends koala.dynamicjava.tree.PrimaryExpression
implements LeftHandSide
```

#### FIELDS

---

- public static final String FIELD\_NAME
  - The body property name
- private String fieldName
  - The field name

#### CONSTRUCTORS

---

- *FieldAccess*

```
protected FieldAccess( java.lang.String fln, java.lang.String fn, int bl,
int bc, int el, int ec )
```

  - **Usage**
    - \* Creates a new field access node
  - **Parameters**
    - \* **fln** - the field name
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

#### METHODS

---

- *getFieldName*

```
public String getFieldName( )
```

  - **Usage**
    - \* Returns the field name
- *setFieldName*

```
public void setFieldName( java.lang.String s )
```

  - **Usage**
    - \* Sets the field name



METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* **listener** - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* **propertyName** - The name of the property to listen on.  
\* **listener** - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*  
 public Object **getProperty**( java.lang.String name )  


---

  - **Usage**  
 \* Returns the value of a property
  - **Parameters**  
 \* name - the property name
  - **Returns** - null if the property was not previously set
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  


---

  - **Usage**  
 \* Returns true if a property is defined for this node
  - **Parameters**  
 \* name - the name of the property
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  


---

  - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**  
 \* listener - The PropertyChangeListener to be removed
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  


---

  - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed
- *setBeginColumn*  
 public void **setBeginColumn**( int i )  


---

  - **Usage**  
 \* Sets the begin column
- *setBeginLine*  
 public void **setBeginLine**( int i )  


---

  - **Usage**  
 \* Sets the begin line
- *setEndColumn*  
 public void **setEndColumn**( int i )  


---

  - **Usage**  
 \* Sets the end column
- *setEndLine*  
 public void **setEndLine**( int i )  


---

  - **Usage**

- \* Sets the end line
- - *setFilename*  
`public void setFilename( java.lang.String s )`
    - **Usage**
      - \* Sets the filename

---
- *setProperty*  
`public void setProperty( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

## 21.2.44 CLASS FieldDeclaration

---

This class represents field declarations in an AST

### DECLARATION

---

```
public class FieldDeclaration
extends koala.dynamicjava.tree.Node
```

### FIELDS

---

- public static final String ACCESS\_FLAGS
  - The accessFlags property name
- public static final String TYPE
  - The type property name
- public static final String NAME
  - The name property name
- public static final String INITIALIZER
  - The initializer property name
- private int accessFlags
  - The access flags
- private Type type
  - The type of this field
- private String name
  - The name of this field
- private Expression initializer
  - The initializer

CONSTRUCTORS

---

• *FieldDeclaration*

```
public FieldDeclaration( int  flags, koala.dynamicjava.tree.Type  type,
    java.lang.String  name, koala.dynamicjava.tree.Expression  init )
```

## – Usage

\* Creates a new field declaration

## – Parameters

\* **flags** - the access flags  
 \* **type** - the type of this field  
 \* **name** - the name of this field  
 \* **init** - the initializer. Can be null

---

• *FieldDeclaration*

```
public FieldDeclaration( int  flags, koala.dynamicjava.tree.Type  type,
    java.lang.String  name, koala.dynamicjava.tree.Expression  init,
    java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

## – Usage

\* Creates a new field declaration

## – Parameters

\* **flags** - the access flags  
 \* **type** - the type of this field  
 \* **name** - the name of this field  
 \* **init** - the initializer. Can be null  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

---

• *getAccessFlags*

```
public int getAccessFlags( )
```

## – Usage

\* Returns the access flags for this method

---

• *getInitializer*

```
public Expression getInitializer( )
```

- **Usage**
    - \* Returns the initializer for this field

---
- *getName*

```
public String getName( )
```

    - **Usage**
      - \* Returns the name of this field

---
  - *getType*

```
public Type getType( )
```

    - **Usage**
      - \* Gets the declared type for this field

---
  - *setAccessFlags*

```
public void setAccessFlags( int f )
```

    - **Usage**
      - \* Sets the access flags for this constructor

---
  - *setInitializer*

```
public void setInitializer( koala.dynamicjava.tree.Expression e )
```

    - **Usage**
      - \* Sets the initializer

---
  - *setName*

```
public void setName( java.lang.String s )
```

    - **Usage**
      - \* Sets the field's name
    - **Exceptions**
      - \* `java.lang.IllegalArgumentException` - if `s` is null

---
  - *setType*

```
public void setType( koala.dynamicjava.tree.Type t )
```

    - **Usage**
      - \* Sets the type of this field
    - **Exceptions**
      - \* `java.lang.IllegalArgumentException` - if `t` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree

- **Parameters**
    - \* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener  
will be invoked only when a call on firePropertyChange names that  
specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue,
int newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName,
java.lang.Object oldValue, java.lang.Object newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.

\* **newValue** - The new value of the property.

- 
- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String **name** )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* **name** - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String **name** )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* **name** - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )  
 – **Usage**



- \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
- **Parameters**
  - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*  

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*  

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.45 CLASS FloatLiteral

---

This class represents the float literal nodes of the syntax tree

DECLARATION

---

```
public class FloatLiteral
extends koala.dynamicjava.tree.Literal
```

CONSTRUCTORS

---

- *FloatLiteral*  

```
public FloatLiteral( java.lang.String rep )
```

  - **Usage**
    - \* Initializes a literal
  - **Parameters**
    - \* **rep** - the representation of the literal

---
- *FloatLiteral*  

```
public FloatLiteral( java.lang.String rep, java.lang.String fn, int bl,
int bc, int el, int ec )
```

  - **Usage**
    - \* Initializes a literal
  - **Parameters**
    - \* **rep** - the representation of the literal
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Literal

---

( in 21.2.67, page 1691)

- *acceptVisitor*  

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

---
- *getRepresentation*  

```
public String getRepresentation( )
```

  - **Usage**
    - \* Returns the representation of this object

---
- *getType*  

```
public Class getType( )
```

  - **Usage**

- \* Returns the type of this expression.  
NOTE: the 'null' literal has a null type

- 
- *getValue*  
public Object **getValue**( )  
    - **Usage**  
\* Returns the value of this expression
  - *setRepresentation*  
public void **setRepresentation**( java.lang.String s )  
    - **Usage**  
\* Sets the representation of this object
    - **Exceptions**  
\* java.lang.IllegalArgumentException - if s is null
  - *setType*  
public void **setType**( java.lang.Class c )  
    - **Usage**  
\* Sets the type of this object
  - *setValue*  
public void **setValue**( java.lang.Object o )  
    - **Usage**  
\* Sets the value of this object
    - **Exceptions**  
\* java.lang.IllegalArgumentException - if o is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
  - **Usage**  
\* Allows a visitor to traverse the tree
  - **Parameters**  
\* visitor - the visitor to accept
- *addPropertyChangeListener*  
public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )

- **Usage**
    - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

- 
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int **i** )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int **i** )
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
 public void **setEndLine**( int **i** )
  - **Usage**
    - \* Sets the end line

---
- *setFilename*  
 public void **setFilename**( java.lang.String **s** )
  - **Usage**
    - \* Sets the filename

---
- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.46 CLASS FloatType

---

This class represents the float type nodes of the syntax tree

### DECLARATION

---

```
public class FloatType
extends koala.dynamicjava.tree.PrimitiveType
```

CONSTRUCTORS

---

• *FloatType***public FloatType( )**– **Usage**

\* Initializes the type

– **Parameters**\* **fn** - the filename\* **bl** - the begin line\* **bc** - the begin column\* **el** - the end line\* **ec** - the end column

---

• *FloatType***public FloatType( java.lang.String fn, int bl, int bc, int el, int ec )**– **Usage**

\* Initializes the type

– **Parameters**\* **fn** - the filename\* **bl** - the begin line\* **bc** - the begin column\* **el** - the end line\* **ec** - the end columnMETHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimitiveType`

---

( in 21.2.89, page 1795)

• *acceptVisitor***public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**\* **visitor** - the visitor to accept

---

• *getValue***public Class getValue( )**– **Usage**\* Returns the value of this node

---

• *setValue***public void setValue( java.lang.Class c )**– **Usage**

\* Sets the value of this node

– **Exceptions**\* `java.lang.IllegalArgumentException` - if `c` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**



- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*
    - public void **setProperty**( java.lang.String name, java.lang.Object value )
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.47 CLASS FormalParameter

---

This class represents the method parameters in the syntax tree

#### DECLARATION

---

```
public class FormalParameter
extends koala.dynamicjava.tree.Node
```

#### FIELDS

---

- public static final String FINAL
  - The final property name
- public static final String TYPE
  - The type property name
- public static final String NAME
  - The name property name
- private boolean finalParameter
  - Is this parameter final?
- private Type type
  - The type of this parameter
- private String name
  - The name of this parameter

CONSTRUCTORS

---

• *FormalParameter*

```
public FormalParameter( boolean f, koala.dynamicjava.tree.Type t,
    java.lang.String n )
```

– **Usage**

\* Initializes the node

– **Parameters**

\* **f** - is the parameter final?  
 \* **t** - the type of the parameter  
 \* **n** - the name of the parameter

---

• *FormalParameter*

```
public FormalParameter( boolean f, koala.dynamicjava.tree.Type t,
    java.lang.String n, java.lang.String fn, int bl, int bc, int el, int ec
    )
```

– **Usage**

\* Initializes the node

– **Parameters**

\* **f** - is the parameter final?  
 \* **t** - the type of the parameter  
 \* **n** - the name of the parameter  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getName*

```
public String getName( )
```

– **Usage**

\* The name of this parameter

---

• *getType*

```
public Type getType( )
```

– **Usage**

- \* Returns the declaring type of this parameter
- 
- *isFinal*  
 public boolean **isFinal**( )  
 – **Usage**  
 \* Is this parameter final?
- 
- *setName*  
 public void **setName**( java.lang.String s )  
 – **Usage**  
 \* Sets this parameter's name  
 – **Exceptions**  
 \* java.lang.IllegalArgumentException - if s is null
- 
- *setType*  
 public void **setType**( koala.dynamicjava.tree.Type t )  
 – **Usage**  
 \* Sets the type of this parameter  
 – **Exceptions**  
 \* java.lang.IllegalArgumentException - if t is null

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
 \* Allows a visitor to traverse the tree  
 – **Parameters**  
 \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*  
**public int getEndLine( )**

- **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  
`public String getFilename( )`
    - **Usage**
      - \* Returns the filename. Can be null.

---
- *getProperties*  
`public Set getProperties( )`
    - **Usage**
      - \* Returns the defined properties for this node.
    - **Returns** - a set of string

---
- *getProperty*  
`public Object getProperty( java.lang.String name )`
    - **Usage**
      - \* Returns the value of a property
    - **Parameters**
      - \* **name** - the property name
    - **Returns** - null if the property was not previously set

---
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`
    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`
    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`
    - **Usage**
      - \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**
      - \* **propertyName** - The name of the property that was listened on.
      - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
`public void setBeginColumn( int i )`
    - **Usage**
      - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
   \* Sets the begin line  


---
- *setEndColumn*  
 public void **setEndColumn**( int i )  
 – **Usage**  
   \* Sets the end column  


---
- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.48 CLASS ForStatement

---

This class represents the for statement nodes of the syntax tree

#### DECLARATION

---

```
public class ForStatement
extends koala.dynamicjava.tree.Statement
implements ContinueTarget
```

#### FIELDS

---

- public static final String INITIALIZATION  
 – The initialization property name
- public static final String CONDITION  
 – The condition property name
- public static final String UPDATE



- The update property name
- public static final String BODY
  - The body property name
- private List initialization
  - The initialization statements
- private Expression condition
  - The condition to evaluate at each loop
- private List update
  - The update statements
- private Node body
  - The body of this statement
- private List labels
  - The labels

## CONSTRUCTORS

---

- *ForStatement*

```
public ForStatement( java.util.List  init, koala.dynamicjava.tree.Expression
cond, java.util.List  updt, koala.dynamicjava.tree.Node  body )
```

  - **Usage**
    - \* Creates a new for statement
  - **Parameters**
    - \* **init** - the initialization statements
    - \* **cond** - the condition to evaluate at each loop
    - \* **updt** - the update statements
    - \* **body** - the body

---
- *ForStatement*

```
public ForStatement( java.util.List  init, koala.dynamicjava.tree.Expression
cond, java.util.List  updt, koala.dynamicjava.tree.Node  body,
java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

  - **Usage**
    - \* Creates a new for statement
  - **Parameters**
    - \* **init** - the initialization statements
    - \* **cond** - the condition to evaluate at each loop
    - \* **updt** - the update statements
    - \* **body** - the body
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *addLabel*

```
public void addLabel( java.lang.String  label )
```

– **Usage**

\* Adds a label to this statement

– **Parameters**

\* **label** - the label to add

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if label is null

---

• *getBody*

```
public Node getBody( )
```

– **Usage**

\* Returns the body of this statement

---

• *getCondition*

```
public Expression getCondition( )
```

– **Usage**

\* Gets the condition to evaluate at each loop

---

• *getInitialization*

```
public List getInitialization( )
```

– **Usage**

\* Gets the initialization statements

---

• *getUpdate*

```
public List getUpdate( )
```

– **Usage**

\* Gets the update statements

---

• *hasLabel*

```
public boolean hasLabel( java.lang.String  label )
```

– **Usage**

\* Test whether this statement has the given label

– **Returns** - true if this statement has the given label

- 
- *setBody*  
 public void **setBody**( koala.dynamicjava.tree.Node node )  
 – **Usage**  
   \* Sets the body of this statement  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if node is null

---

  - *setCondition*  
 public void **setCondition**( koala.dynamicjava.tree.Expression e )  
 – **Usage**  
   \* Sets the condition to evaluate

---

  - *setInitialization*  
 public void **setInitialization**( java.util.List l )  
 – **Usage**  
   \* Sets the initialization statements

---

  - *setUpdate*  
 public void **setUpdate**( java.util.List l )  
 – **Usage**  
   \* Sets the update statements

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Statement

---

( in 21.2.102, page 1857)

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* visitor - the visitor to accept

---

- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener to the listener list.  
   The listener is registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

- 
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.

- **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.49 CLASS FunctionCall

---

This class represents the function call nodes of the syntax tree

### DECLARATION

---

```
public class FunctionCall
extends koala.dynamicjava.tree.MethodCall
```

CONSTRUCTORS

---

- *FunctionCall*

```
public FunctionCall( java.lang.String mn, java.util.List args,
    java.lang.String fn, int bl, int bc, int el, int ec )
```

- **Usage**

- \* Creates a new node

- **Parameters**

- \* **mn** - the function name
    - \* **args** - the arguments. Can be null.
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.MethodCall

---

( in 21.2.70, page 1706)

- *getArguments*

```
public List getArguments( )
```

- **Usage**

- \* Returns the arguments.

- **Returns** - null if there is no argument

---

- *getMethodName*

```
public String getMethodName( )
```

- **Usage**

- \* Returns the method name

---

- *setArguments*

```
public void setArguments( java.util.List l )
```

- **Usage**

- \* Sets the constructor arguments.

---

- *setMethodName*

```
public void setMethodName( java.lang.String s )
```

- **Usage**
  - \* Sets the method name
- **Exceptions**
  - \* `java.lang.IllegalArgumentException` - if `s` is null

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added
- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )`
  - **Usage**



- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.
-

- *getProperties*  
 public Set **getProperties**( )  


---

  - **Usage**  
 \* Returns the defined properties for this node.
  - **Returns** - a set of string
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  


---

  - **Usage**  
 \* Returns the value of a property
  - **Parameters**  
 \* name - the property name
  - **Returns** - null if the property was not previously set
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  


---

  - **Usage**  
 \* Returns true if a property is defined for this node
  - **Parameters**  
 \* name - the name of the property
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  


---

  - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**  
 \* listener - The PropertyChangeListener to be removed
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  


---

  - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed
- *setBeginColumn*  
 public void **setBeginColumn**( int i )  


---

  - **Usage**  
 \* Sets the begin column
- *setBeginLine*  
 public void **setBeginLine**( int i )  


---

  - **Usage**  
 \* Sets the begin line
- *setEndColumn*  
 public void **setEndColumn**( int i )

- **Usage**
    - \* Sets the end column
- 
- *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- 
- *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.50 CLASS GreaterExpression

This class represents the greater expression nodes of the syntax tree

#### DECLARATION

```
public class GreaterExpression
extends koala.dynamicjava.tree.BinaryExpression
```

#### CONSTRUCTORS

- *GreaterExpression*

```
public GreaterExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **lexp** - the LHS expression
      - \* **rexp** - the RHS expression
- 
- *GreaterExpression*

```
public GreaterExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

    - **Usage**

- \* Initializes the expression
- **Parameters**
  - \* **lexp** - the LHS expression
  - \* **rexp** - the RHS expression
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor **visitor** )
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*  
 public Expression **getLeftExpression**( )
  - **Usage**
    - \* Returns the left hand side expression
- *getRightExpression*  
 public Expression **getRightExpression**( )
  - **Usage**
    - \* Returns the right hand side expression
- *setLeftExpression*  
 public void **setLeftExpression**( koala.dynamicjava.tree.Expression **exp** )
  - **Usage**
    - \* Sets the left hand side expression
  - **Exceptions**
    - \* java.lang.IllegalArgumentException - if exp is null
- *setRightExpression*  
 public void **setRightExpression**( koala.dynamicjava.tree.Expression **exp** )
  - **Usage**
    - \* Sets the right hand side expression
  - **Exceptions**
    - \* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String **name** )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**, java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int **i** )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int **i** )
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
 public void **setEndLine**( int **i** )
  - **Usage**
    - \* Sets the end line

---
- *setFilename*  
 public void **setFilename**( java.lang.String **s** )
  - **Usage**

- \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.51 CLASS GreaterOrEqualExpression

---

This class represents the greater or equal expression nodes of the syntax tree

#### DECLARATION

---

```
public class GreaterOrEqualExpression
extends koala.dynamicjava.tree.BinaryExpression
```

#### CONSTRUCTORS

---

- *GreaterOrEqualExpression*

```
public GreaterOrEqualExpression( koala.dynamicjava.tree.Expression  lexp,
koala.dynamicjava.tree.Expression  rexp )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **lexp** - the LHS expression
      - \* **rexp** - the RHS expression
- 
- *GreaterOrEqualExpression*

```
public GreaterOrEqualExpression( koala.dynamicjava.tree.Expression  lexp,
koala.dynamicjava.tree.Expression  rexp, java.lang.String  fn, int  bl, int
bc, int  el, int  ec )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **lexp** - the LHS expression
      - \* **rexp** - the RHS expression
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column



METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

– **Usage**

\* Returns the left hand side expression

• *getRightExpression*

```
public Expression getRightExpression( )
```

– **Usage**

\* Returns the right hand side expression

• *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression  exp )
```

– **Usage**

\* Sets the left hand side expression

– **Exceptions**

\* java.lang.IllegalArgumentException - if exp is null

• *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression  exp )
```

– **Usage**

\* Sets the right hand side expression

– **Exceptions**

\* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.52 CLASS Identifier

---

This class implements a tree token

#### DECLARATION

---

```
public class Identifier
extends java.lang.Object
implements IdentifierToken
```

#### FIELDS

---

- private String image
  - The image
- private int beginLine
  - The begin line
- private int beginColumn
  - The begin line
- private int endLine
  - The end line
- private int endColumn
  - The end line

#### CONSTRUCTORS

---

- *Identifier*  

```
public Identifier( java.lang.String im )
```

  - **Usage**
    - \* Creates a new token
  - **Parameters**
    - \* im - the image

---
- *Identifier*  

```
public Identifier( java.lang.String im, int bl, int bc, int el, int ec )
```

  - **Usage**
    - \* Creates a new token
  - **Parameters**
    - \* im - the image
    - \* bl - the begin line
    - \* bc - the begin column
    - \* el - the end line
    - \* ec - the end column

METHODS

---

- *beginColumn*  
**public int beginColumn( )**
  - **Usage**
    - \* Returns the column number where the beginning of the token was found in the source file

---
- *beginLine*  
**public int beginLine( )**
  - **Usage**
    - \* Returns the line number where the beginning of the token was found in the source file

---
- *endColumn*  
**public int endColumn( )**
  - **Usage**
    - \* Returns the column number where the end of the token was found in the source file

---
- *endLine*  
**public int endLine( )**
  - **Usage**
    - \* Returns the line number where the end of the token was found in the source file

---
- *image*  
**public String image( )**
  - **Usage**
    - \* Returns the representation of the identifier

**21.2.53 CLASS IfThenElseStatement**

---

This class represents the if-then-else statement nodes of the syntax tree

DECLARATION

---

```
public class IfThenElseStatement
extends koala.dynamicjava.tree.IfThenStatement
```

FIELDS

---

- public static final String ELSE\_STATEMENT
  - The elseStatement property name
- private Node elseStatement
  - The 'else' statement

CONSTRUCTORS

---

- *IfThenElseStatement*  
public **IfThenElseStatement**( koala.dynamicjava.tree.Expression cond,  
koala.dynamicjava.tree.Node tstmt, koala.dynamicjava.tree.Node estmt )
    - Usage
      - \* Creates a new while statement
    - Parameters
      - \* cond - the condition
      - \* tstmt - the then statement
      - \* estmt - the else statement
- 
- *IfThenElseStatement*  
public **IfThenElseStatement**( koala.dynamicjava.tree.Expression cond,  
koala.dynamicjava.tree.Node tstmt, koala.dynamicjava.tree.Node estmt,  
java.lang.String fn, int bl, int bc, int el, int ec )
    - Usage
      - \* Creates a new while statement
    - Parameters
      - \* cond - the condition
      - \* tstmt - the then statement
      - \* estmt - the else statement
      - \* fn - the filename
      - \* bl - the begin line
      - \* bc - the begin column
      - \* el - the end line
      - \* ec - the end column

METHODS

---

- *acceptVisitor*  
public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
  - Usage
    - \* Allows a visitor to traverse the tree
  - Parameters
    - \* visitor - the visitor to accept

---

- *getElseStatement*

public Node **getElseStatement**( )

- **Usage**

- \* Returns the else statement of this statement

---

- *setElseStatement*

public void **setElseStatement**( koala.dynamicjava.tree.Node node )

- **Usage**

- \* Sets the else statement of this statement

- **Exceptions**

- \* java.lang.IllegalArgumentException - if node is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.IfThenStatement

---

( in 21.2.54, page 1627)

- *acceptVisitor*

public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* visitor - the visitor to accept

---

- *getCondition*

public Expression **getCondition**( )

- **Usage**

- \* Gets the condition to evaluate at each loop

---

- *getThenStatement*

public Node **getThenStatement**( )

- **Usage**

- \* Returns the then statement of this statement

---

- *setCondition*

public void **setCondition**( koala.dynamicjava.tree.Expression e )

- **Usage**

- \* Sets the condition to evaluate

- **Exceptions**

- \* java.lang.IllegalArgumentException - if e is null

---

- *setThenStatement*

public void **setThenStatement**( koala.dynamicjava.tree.Node node )

- **Usage**

- \* Sets the then statement of this statement

- **Exceptions**

- \* java.lang.IllegalArgumentException - if node is null



METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
- **Parameters**
  - \* **propertyName** - The programmatic name of the property that was changed.
  - \* **oldValue** - The old value of the property.
  - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*  
**public int getEndLine( )**
  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*  
**public String getFilename( )**
  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*  
**public Set getProperties( )**
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---

- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.54 CLASS IfThenStatement

---

This class represents the if-then statement nodes of the syntax tree

#### DECLARATION

---

```
public class IfThenStatement
extends koala.dynamicjava.tree.Statement
```

#### FIELDS

---

- public static final String CONDITION
  - The condition property name
- public static final String THEN\_STATEMENT
  - The thenStatement property name
- private Expression condition
  - The condition
- private Node thenStatement
  - The then-statement of this statement

#### CONSTRUCTORS

---

- *IfThenStatement*

```
public IfThenStatement( koala.dynamicjava.tree.Expression  cond,
koala.dynamicjava.tree.Node  tstmt )
```

    - **Usage**
      - \* Creates a new while statement
    - **Parameters**
      - \* **cond** - the condition
      - \* **tstmt** - the statement
-

- *IfThenStatement*

```
public IfThenStatement( koala.dynamicjava.tree.Expression cond,
koala.dynamicjava.tree.Node tstmt, java.lang.String fn, int bl, int bc,
int el, int ec )
```

- **Usage**

- \* Creates a new while statement

- **Parameters**

- \* **cond** - the condition
    - \* **tstmt** - the statement
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getCondition*

```
public Expression getCondition( )
```

- **Usage**

- \* Gets the condition to evaluate at each loop

---

- *getThenStatement*

```
public Node getThenStatement( )
```

- **Usage**

- \* Returns the then statement of this statement

---

- *setCondition*

```
public void setCondition( koala.dynamicjava.tree.Expression e )
```

- **Usage**

- \* Sets the condition to evaluate

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if e is null

---

- *setThenStatement*

```
public void setThenStatement( koala.dynamicjava.tree.Node node )
```

- **Usage**

- \* Sets the then statement of this statement

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if node is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`

---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**



- \* Sets the filename
- 
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.55 CLASS ImportDeclaration

---

This class represents the import declarations

#### DECLARATION

---

```
public class ImportDeclaration
extends koala.dynamicjava.tree.Node
```

#### FIELDS

---

- public static final String NAME
  - The name property name
- public static final String PACKAGE
  - The package property name
- private String name
  - The name of the imported class or package
- private boolean pckage
  - Is this declaration import a class or a package

#### CONSTRUCTORS

---

- *ImportDeclaration*  
 public **ImportDeclaration**( java.util.List ident, boolean pkg )
    - **Usage**
      - \* Creates a new import declaration node
    - **Parameters**
      - \* **ident** - a list of tokens that represents a package or a class name
      - \* **pkg** - true if this declaration imports a package
-

- *ImportDeclaration*

```
public ImportDeclaration( java.util.List ident, boolean pkg,
    java.lang.String fn, int bl, int bc, int el, int ec )
```

- **Usage**

- \* Creates a new import declaration node

- **Parameters**

- \* **ident** - a list of tokens that represents a package or a class name
    - \* **pkg** - true if this declaration imports a package
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getName*

```
public String getName( )
```

- **Usage**

- \* Returns the name of the imported class or package

---

- *isPackage*

```
public boolean isPackage( )
```

- **Usage**

- \* Returns true if the identifier represents a package, false if it represents a

---

- *setName*

```
public void setName( java.lang.String s )
```

- **Usage**

- \* Sets the package name

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if s is null

---

- *setPackage*

```
public void setPackage( boolean b )
```

- **Usage**

- \* Sets the package property

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.56 CLASS **Initializer**

---

This class represents the initializer statement nodes of the syntax tree

#### DECLARATION

---

```
public abstract class Initializer
extends koala.dynamicjava.tree.Node
```

#### FIELDS

---

- public static final String BLOCK
  - The block property name
- private BlockStatement block
  - The block

#### CONSTRUCTORS

---

- *Initializer*  

```
protected Initializer( koala.dynamicjava.tree.BlockStatement  block,
java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

  - **Usage**
    - \* Creates a new initializer statement
  - **Parameters**
    - \* **block** - the block
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

#### METHODS

---

- *getBlock*  

```
public BlockStatement getBlock( )
```

  - **Usage**
    - \* Gets the block statement
- *setBlock*  

```
public void setBlock( koala.dynamicjava.tree.BlockStatement  bs )
```

  - **Usage**
    - \* Sets the block statement
  - **Exceptions**
    - \* java.lang.IllegalArgumentException - if bs is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
    java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**



- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.57 CLASS InnerAllocation

---

This class represents the inner allocation nodes of the syntax tree

### DECLARATION

---

```
public class InnerAllocation
extends koala.dynamicjava.tree.Allocation
implements ExpressionStatement, ExpressionContainer
```

### FIELDS

---

- public static final String ARGUMENTS
  - The arguments property name
- private Expression expression
  - The outer object expression
- private List arguments
  - The arguments to pass to the constructor

### CONSTRUCTORS

---

- *InnerAllocation*

```
public InnerAllocation( koala.dynamicjava.tree.Expression exp,
koala.dynamicjava.tree.Type tp, java.util.List args )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the outer object
    - \* **tp** - the type prefix
    - \* **args** - the arguments of the constructor. null if no arguments.

---
- *InnerAllocation*

```
public InnerAllocation( koala.dynamicjava.tree.Expression exp,
koala.dynamicjava.tree.Type tp, java.util.List args, java.lang.String fn,
int bl, int bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the outer object
    - \* **tp** - the type prefix
    - \* **args** - the arguments of the constructor. null if no arguments.
    - \* **fn** - the filename

\* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

\* Allows a visitor to traverse the tree

- **Parameters**

\* **visitor** - the visitor to accept

---

- *getArguments*

```
public List getArguments( )
```

- **Usage**

\* Returns the constructor arguments.

- **Returns** - null if there is no argument.

---

- *getExpression*

```
public Expression getExpression( )
```

- **Usage**

\* Returns the outer class instance expression

---

- *setArguments*

```
public void setArguments( java.util.List  l )
```

- **Usage**

\* Sets the constructor arguments.

---

- *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression  e )
```

- **Usage**

\* Sets the outer class instance expression

- **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

## METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Allocation`

---

( in 21.2.3, page 1387)

- *getCreationType*

```
public Type getCreationType( )
```

- **Usage**

\* Returns the creation type

- 
- *setCreationType*  
 public void **setCreationType**( koala.dynamicjava.tree.Type t )  
 – **Usage**  
   \* Sets the creation type  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if t is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* visitor - the visitor to accept
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener to the listener list.  
   The listener is registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be added
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.  
 – **Parameters**  
   \* propertyName - The name of the property to listen on.  
   \* listener - The PropertyChangeListener to be added
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String propertyName, boolean oldValue, boolean newValue )

- **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *getBeginColumn*  
public int **getBeginColumn**( )
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  
public int **getBeginLine**( )
  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  
public int **getEndColumn**( )
  - **Usage**
    - \* Returns the end column of this node in the end line

---
- *getEndLine*  
public int **getEndLine**( )
  - **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  
public String **getFilename**( )
  - **Usage**
    - \* Returns the filename. Can be null.

---

- 
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
   \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns the value of a property  
 – **Parameters**  
   \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns true if a property is defined for this node  
 – **Parameters**  
   \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener from the listener list.  
   This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* propertyName - The name of the property that was listened on.  
   \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
   \* Sets the begin column

---

  - *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
   \* Sets the begin line

---

  - *setEndColumn*  
 public void **setEndColumn**( int i )

- **Usage**
  - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.58 CLASS InnerClassAllocation

---

This class represents the anonymous inner classe allocation nodes of the syntax tree

#### DECLARATION

---

```
public class InnerClassAllocation
extends koala.dynamicjava.tree.InnerAllocation
```

#### FIELDS

---

- public static final String MEMBERS
  - The members property name
- private List members
  - The members of the anonymous class

#### CONSTRUCTORS

---

- *InnerClassAllocation*

```
public InnerClassAllocation( koala.dynamicjava.tree.Expression exp,
koala.dynamicjava.tree.Type tp, java.util.List args, java.util.List memb
)
```

  - **Usage**

- \* Initializes the expression
  - **Parameters**
    - \* **exp** - the outer object
    - \* **tp** - the type prefix
    - \* **args** - the arguments of the constructor. Can be null.
    - \* **memb** - the members of the class
- 
- *InnerClassAllocation*

```
public InnerClassAllocation( koala.dynamicjava.tree.Expression exp,
koala.dynamicjava.tree.Type tp, java.util.List args, java.util.List
memb, java.lang.String fn, int bl, int bc, int el, int ec )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **exp** - the outer object
      - \* **tp** - the type prefix
      - \* **args** - the arguments of the constructor. Can be null.
      - \* **memb** - the members of the class
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* **visitor** - the visitor to accept
- 
- *getMembers*

```
public List getMembers( )
```

    - **Usage**
      - \* Returns the members of the anonymous class
- 
- *setMembers*

```
public void setMembers( java.util.List l )
```

    - **Usage**
      - \* Sets the members of the anonymous class
    - **Exceptions**
      - \* `java.lang.IllegalArgumentException` - if `l` is null



METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.InnerAllocation`

---

( in 21.2.57, page 1641)

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *getArguments*  
`public List getArguments( )`
  - **Usage**
    - \* Returns the constructor arguments.
  - **Returns** - null if there is no argument.
- *getExpression*  
`public Expression getExpression( )`
  - **Usage**
    - \* Returns the outer class instance expression
- *setArguments*  
`public void setArguments( java.util.List l )`
  - **Usage**
    - \* Sets the constructor arguments.
- *setExpression*  
`public void setExpression( koala.dynamicjava.tree.Expression e )`
  - **Usage**
    - \* Sets the outer class instance expression
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if `e` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Allocation`

---

( in 21.2.3, page 1387)

- *getCreationType*  
`public Type getCreationType( )`
  - **Usage**
    - \* Returns the creation type
- *setCreationType*  
`public void setCreationType( koala.dynamicjava.tree.Type t )`
  - **Usage**
    - \* Sets the creation type
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if `t` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* **listener** - The PropertyChangeListener to be added• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener  
will be invoked only when a call on firePropertyChange names that  
specific property.

## – Parameters

\* **propertyName** - The name of the property to listen on.  
\* **listener** - The PropertyChangeListener to be added• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
**public void setEndColumn( int i )**
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
**public void setEndLine( int i )**
  - **Usage**

- \* Sets the end line
- - *setFilename*  
`public void setFilename( java.lang.String s )`
    - **Usage**
      - \* Sets the filename

---
- *setProperty*  
`public void setProperty( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.59 CLASS InstanceInitializer

---

This class represents the instance initializer statement nodes of the syntax tree

#### DECLARATION

---

```
public class InstanceInitializer
extends koala.dynamicjava.tree.Initializer
```

#### CONSTRUCTORS

---

- *InstanceInitializer*  
`public InstanceInitializer( koala.dynamicjava.tree.BlockStatement block )`
    - **Usage**
      - \* Creates a new initializer statement
    - **Parameters**
      - \* **block** - the block

---
- *InstanceInitializer*  
`public InstanceInitializer( koala.dynamicjava.tree.BlockStatement block, java.lang.String fn, int bl, int bc, int el, int ec )`
    - **Usage**
      - \* Creates a new initializer statement
    - **Parameters**
      - \* **block** - the block
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Initializer

( in 21.2.56, page 1637)

- *getBlock*

```
public BlockStatement getBlock( )
```

- **Usage**

- \* Gets the block statement

---

- *setBlock*

```
public void setBlock( koala.dynamicjava.tree.BlockStatement  bs )
```

- **Usage**

- \* Sets the block statement

- **Exceptions**

- \* java.lang.IllegalArgumentException - if bs is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName, java.beans.PropertyChangeListener  listener )
```

- **Usage**
  - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
- **Parameters**
  - \* **propertyName** - The name of the property to listen on.
  - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**

- 
- \* Returns the end column of this node in the end line
- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code
- 
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.
- 
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string
- 
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set
- 
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property
- 
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed
-



- *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
   \* Sets the begin column  


---
- *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
   \* Sets the begin line  


---
- *setEndColumn*  
 public void **setEndColumn**( int i )  
 – **Usage**  
   \* Sets the end column  


---
- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.60 CLASS InstanceOfExpression

---

This class represents the instanceof expression nodes of the syntax tree

#### DECLARATION

---

```
public class InstanceOfExpression
extends koala.dynamicjava.tree.Expression
implements ExpressionContainer
```

FIELDS

---

- public static final String REFERENCE\_TYPE
  - The referenceType property name
- private Expression expression
  - The expression to check
- private Type referenceType
  - The type to check

CONSTRUCTORS

---

- *InstanceOfExpression*

```
public InstanceOfExpression( koala.dynamicjava.tree.Expression exp,
                             koala.dynamicjava.tree.Type t )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **exp** - the expression to test
      - \* **t** - the type to check
- 
- *InstanceOfExpression*

```
public InstanceOfExpression( koala.dynamicjava.tree.Expression exp,
                             koala.dynamicjava.tree.Type t, java.lang.String fn, int bl, int bc, int
                             el, int ec )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **exp** - the expression to test
      - \* **t** - the type to check
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**

- \* **visitor** - the visitor to accept

---

  - *getExpression*  
**public Expression getExpression( )**
    - **Usage**
      - \* Returns the expression to check

---

- *getReferenceType*  
**public Type getReferenceType( )**
  - **Usage**
    - \* Returns the type to check

---

- *setExpression*  
**public void setExpression( koala.dynamicjava.tree.Expression e )**
  - **Usage**
    - \* Sets the expression to check
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if e is null

---

- *setReferenceType*  
**public void setReferenceType( koala.dynamicjava.tree.Type t )**
  - **Usage**
    - \* Sets the type to check
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if t is null

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
**public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*  
**public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )**
  - **Usage**

- \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code
 

---
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line
 

---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code
 

---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.
 

---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string
 

---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set
 

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property
 

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed
 

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* **propertyName** - The name of the property that was listened on.  
   \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
   \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
   \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
   \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
   \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
   \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.61 CLASS IntegerLiteral

---

This class represents the integer literal nodes of the syntax tree

#### DECLARATION

---

```
public class IntegerLiteral
extends koala.dynamicjava.tree.Literal
```

CONSTRUCTORS

---

• *IntegerLiteral*

```
public IntegerLiteral( java.lang.String rep )
```

– **Usage**

- \* Initializes a literal

– **Parameters**

- \* **rep** - the representation of the literal

---

• *IntegerLiteral*

```
public IntegerLiteral( java.lang.String rep, java.lang.String fn, int bl,  
int bc, int el, int ec )
```

– **Usage**

- \* Initializes a literal

– **Parameters**

- \* **rep** - the representation of the literal
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

METHODS

---

• *parse*

```
private static Integer parse( java.lang.String s )
```

– **Usage**

- \* Parses the representation of an integer

---

• *parseHexadecimal*

```
private static Integer parseHexadecimal( java.lang.String s )
```

– **Usage**

- \* Parses an hexadecimal number

---

• *parseOctal*

```
private static Integer parseOctal( java.lang.String s )
```

– **Usage**

- \* Parses an octal number

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Literal`

( in 21.2.67, page 1691)

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**  
\* Allows a visitor to traverse the tree
  - **Parameters**  
\* `visitor` - the visitor to accept
- *getRepresentation*  
`public String getRepresentation( )`
  - **Usage**  
\* Returns the representation of this object
- *getType*  
`public Class getType( )`
  - **Usage**  
\* Returns the type of this expression.  
NOTE: the 'null' literal has a null type
- *getValue*  
`public Object getValue( )`
  - **Usage**  
\* Returns the value of this expression
- *setRepresentation*  
`public void setRepresentation( java.lang.String s )`
  - **Usage**  
\* Sets the representation of this object
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if `s` is null
- *setType*  
`public void setType( java.lang.Class c )`
  - **Usage**  
\* Sets the type of this object
- *setValue*  
`public void setValue( java.lang.Object o )`
  - **Usage**  
\* Sets the value of this object
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if `o` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

( in 21.2.88, page 1792)



METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.62 CLASS InterfaceDeclaration

---

This class represents an interface declaration

#### DECLARATION

---

```
public class InterfaceDeclaration
extends koala.dynamicjava.tree.TypeDeclaration
```

#### CONSTRUCTORS

---

- *InterfaceDeclaration*

```
public InterfaceDeclaration( int  flags, java.lang.String  name,
java.util.List  impl, java.util.List  body )
```

    - **Usage**
      - \* Creates a new interface declaration
    - **Parameters**
      - \* **flags** - the access flags
      - \* **name** - the name of the interface to declare
      - \* **impl** - the list of implemented interfaces. Can be null.
      - \* **body** - the list of fields declarations
- 
- *InterfaceDeclaration*

```
public InterfaceDeclaration( int  flags, java.lang.String  name,
java.util.List  impl, java.util.List  body, java.lang.String  fn, int  bl,
int  bc, int  el, int  ec )
```

    - **Usage**
      - \* Creates a new interface declaration
    - **Parameters**
      - \* **flags** - the access flags
      - \* **name** - the name of the interface to declare
      - \* **impl** - the list of implemented interfaces. Can be null.
      - \* **body** - the list of fields declarations
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.TypeDeclaration`

---

( in 21.2.118, page 1928)

• *getAccessFlags*

```
public int getAccessFlags( )
```

– **Usage**

\* Returns the access flags for this class

• *getInterfaces*

```
public List getInterfaces( )
```

– **Usage**

\* Returns a list that contains the names (String) of the implemented interfaces.  
Can be null.

• *getMembers*

```
public List getMembers( )
```

– **Usage**

\* Returns the list of the declared members

• *getName*

```
public String getName( )
```

– **Usage**

\* Returns the name of this class

• *setAccessFlags*

```
public void setAccessFlags( int  f )
```

– **Usage**

\* Sets the access flags for this constructor

• *setInterfaces*

```
public void setInterfaces( java.util.List  l )
```

– **Usage**

\* Sets the interfaces (a list of strings)

• *setMembers*

```
public void setMembers( java.util.List  l )
```

– **Usage**

\* Sets the members

– **Exceptions**

- 
- \* java.lang.IllegalArgumentException - if l is null
- 
- *setName*  
 public void setName( java.lang.String s )
    - **Usage**
      - \* Sets the type's name
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if s is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*  
 public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
 public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.
      - \* listener - The PropertyChangeListener to be added
- 
- *firePropertyChange*  
 protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
    - **Usage**
      - \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.
    - **Parameters**
      - \* propertyName - The programmatic name of the property that was changed.
      - \* oldValue - The old value of the property.
      - \* newValue - The new value of the property.
-

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
**public void setEndColumn( int i )**
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
**public void setEndLine( int i )**
  - **Usage**



- \* Sets the end line
- - *setFilename*  
`public void setFilename( java.lang.String s )`
    - **Usage**
      - \* Sets the filename

---
- *setProperty*  
`public void setProperty( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.63 CLASS IntType

---

This class represents the int type nodes of the syntax tree

#### DECLARATION

---

```
public class IntType
extends koala.dynamicjava.tree.PrimitiveType
```

#### CONSTRUCTORS

---

- *IntType*  
`public IntType( )`
    - **Usage**
      - \* Initializes the type

---
- *IntType*  
`public IntType( java.lang.String fn, int bl, int bc, int el, int ec )`
    - **Usage**
      - \* Initializes the type
    - **Parameters**
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimitiveType`

---

( in 21.2.89, page 1795)

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept

---
- *getValue*  
`public Class getValue( )`
  - **Usage**
    - \* Returns the value of this node

---
- *setValue*  
`public void setValue( java.lang.Class c )`
  - **Usage**
    - \* Sets the value of this node
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if `c` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept

---
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added

---
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`

- **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, boolean **oldValue**, boolean **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
public int **getBeginColumn**( )
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
public int **getBeginLine**( )
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
public int **getEndColumn**( )
  - **Usage**

- 
- \* Returns the end column of this node in the end line
- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code
- 
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.
- 
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string
- 
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set
- 
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property
- 
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed
-

- *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
   \* Sets the begin column  


---
- *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
   \* Sets the begin line  


---
- *setEndColumn*  
 public void **setEndColumn**( int i )  
 – **Usage**  
   \* Sets the end column  


---
- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.64 CLASS LabeledStatement

---

This class represents the labeled statement nodes of the syntax tree

#### DECLARATION

---

```
public class LabeledStatement
extends koala.dynamicjava.tree.Statement
```

FIELDS

---

- public static final String LABEL
  - The label property name
- public static final String STATEMENT
  - The statement property name
- private String label
  - The label
- private Node statement
  - The statement

CONSTRUCTORS

---

- *LabeledStatement*  
 public **LabeledStatement**( java.lang.String label,  
 koala.dynamicjava.tree.Node stat )  
    - **Usage**
      - \* Creates a new while statement
    - **Parameters**
      - \* **label** - the label
      - \* **stat** - the statement
- 
- *LabeledStatement*  
 public **LabeledStatement**( java.lang.String label,  
 koala.dynamicjava.tree.Node stat, java.lang.String fn, int bl, int bc,  
 int el, int ec )  
    - **Usage**
      - \* Creates a new while statement
    - **Parameters**
      - \* **label** - the label
      - \* **stat** - the statement
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )

- **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept
- 
- *getLabel*  
 public String **getLabel**( )
    - **Usage**
      - \* Gets the label
- 
- *getStatement*  
 public Node **getStatement**( )
    - **Usage**
      - \* Returns the statement
- 
- *setLabel*  
 public void **setLabel**( java.lang.String s )
    - **Usage**
      - \* Sets the label
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if e is null
- 
- *setStatement*  
 public void **setStatement**( koala.dynamicjava.tree.Node n )
    - **Usage**
      - \* Sets the statement
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if n is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Statement

---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* **visitor** - the visitor to accept
-

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.
    - The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added
- 

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
-



- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line  


---
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code  


---
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line  


---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code  


---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.  


---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string  


---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set  


---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.

- **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.65 CLASS LessExpression

---

This class represents the less expression nodes of the syntax tree

DECLARATION

---

```
public class LessExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *LessExpression*

```
public LessExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

## – Usage

- \* Initializes the expression

## – Parameters

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression

• *LessExpression*

```
public LessExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

## – Usage

- \* Initializes the expression

## – Parameters

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

- \* Allows a visitor to traverse the tree

## – Parameters

- \* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.BinaryExpression`

---

( in 21.2.11, page 1421)

- *getLeftExpression*  
`public Expression getLeftExpression( )`  
 – **Usage**  
   \* Returns the left hand side expression

---

- *getRightExpression*  
`public Expression getRightExpression( )`  
 – **Usage**  
   \* Returns the right hand side expression

---

- *setLeftExpression*  
`public void setLeftExpression( koala.dynamicjava.tree.Expression exp )`  
 – **Usage**  
   \* Sets the left hand side expression  
 – **Exceptions**  
   \* `java.lang.IllegalArgumentException` - if exp is null

---

- *setRightExpression*  
`public void setRightExpression( koala.dynamicjava.tree.Expression exp )`  
 – **Usage**  
   \* Sets the right hand side expression  
 – **Exceptions**  
   \* `java.lang.IllegalArgumentException` - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`  
 – **Usage**  
   \* Adds a `PropertyChangeListener` to the listener list.  
   The listener is registered for all properties.  
 – **Parameters**

\* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue,
int newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName,
java.lang.Object oldValue, java.lang.Object newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- • *getEndColumn*  
 public int **getEndColumn**( )
  - **Usage**
    - \* Returns the end column of this node in the end line
- • *getEndLine*  
 public int **getEndLine**( )
  - **Usage**
    - \* Returns the end line of this node in the source code
- • *getFilename*  
 public String **getFilename**( )
  - **Usage**
    - \* Returns the filename. Can be null.
- • *getProperties*  
 public Set **getProperties**( )
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string
- • *getProperty*  
 public Object **getProperty**( java.lang.String name )
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* name - the property name
  - **Returns** - null if the property was not previously set
- • *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* name - the name of the property
- • *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* listener - The PropertyChangeListener to be removed
- • *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.

- **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.66 CLASS LessOrEqualExpression

---

This class represents the less or equal expression nodes of the syntax tree

#### DECLARATION

---

```
public class LessOrEqualExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *LessOrEqualExpression*

```
public LessOrEqualExpression( koala.dynamicjava.tree.Expression lexp,
                             koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

• *LessOrEqualExpression*

```
public LessOrEqualExpression( koala.dynamicjava.tree.Expression lexp,
                             koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
                             bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

\* **fn** - the filename

\* **bl** - the begin line

\* **bc** - the begin column

\* **el** - the end line

\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

## – Usage

\* Returns the left hand side expression

• *getRightExpression*

```
public Expression getRightExpression( )
```



- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.

\* **listener** - The PropertyChangeListener to be added

---

• *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**

– **Usage**

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

---

• *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**

– **Usage**

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

---

• *firePropertyChange*

**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**

– **Usage**

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

---

• *getBeginColumn*

**public int getBeginColumn( )**

– **Usage**

\* Returns the begin column of this node in the begin line

---

• *getBeginLine*

**public int getBeginLine( )**

– **Usage**

\* Returns the begin line of this node in the source code

---

• *getEndColumn*

**public int getEndColumn( )**

– **Usage**

\* Returns the end column of this node in the end line

---

• *getEndLine*

**public int getEndLine( )**

– **Usage**

\* Returns the end line of this node in the source code

- 
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
 \* Sets the begin column

---

  - *setBeginLine*  
 public void **setBeginLine**( int i )

- **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.67 CLASS Literal

---

This class represents the literal nodes of the syntax tree

#### DECLARATION

---

```
public abstract class Literal
extends koala.dynamicjava.tree.PrimaryExpression
```

#### FIELDS

---

- public static final String REPRESENTATION
  - The representation property name
- public static final String VALUE
  - The value property name
- public static final String TYPE
  - The type property name
- private String representation
  - The representation of the literal

- private Object value
  - The value of this literal
- private Class type
  - The type of this literal

## CONSTRUCTORS

---

- *Literal*  
`protected Literal( java.lang.String rep, java.lang.Object val,  
 java.lang.Class typ, java.lang.String fn, int bl, int bc, int el, int  
 ec )`
  - **Usage**
    - \* Initializes a literal
  - **Parameters**
    - \* **rep** - the representation of the literal
    - \* **val** - the value of this literal
    - \* **typ** - the type of this literal
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* **visitor** - the visitor to accept
  - *getRepresentation*  
`public String getRepresentation( )`
    - **Usage**
      - \* Returns the representation of this object
  - *getType*  
`public Class getType( )`
    - **Usage**
      - \* Returns the type of this expression.  
 NOTE: the 'null' literal has a null type
-

- *getValue*  
`public Object getValue( )`  
  - **Usage**  
 \* Returns the value of this expression

---
- *setRepresentation*  
`public void setRepresentation( java.lang.String s )`  
  - **Usage**  
 \* Sets the representation of this object
  - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if s is null

---
- *setType*  
`public void setType( java.lang.Class c )`  
  - **Usage**  
 \* Sets the type of this object

---
- *setValue*  
`public void setValue( java.lang.Object o )`  
  - **Usage**  
 \* Sets the value of this object
  - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if o is null

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
  - **Usage**  
 \* Allows a visitor to traverse the tree
  - **Parameters**  
 \* `visitor` - the visitor to accept

---
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`

- **Usage**
  - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
- **Parameters**
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

- 
- *getBeginLine*  
`public int getBeginLine( )`  
    - **Usage**  
 \* Returns the begin line of this node in the source code
- 
- *getEndColumn*  
`public int getEndColumn( )`  
    - **Usage**  
 \* Returns the end column of this node in the end line
- 
- *getEndLine*  
`public int getEndLine( )`  
    - **Usage**  
 \* Returns the end line of this node in the source code
- 
- *getFilename*  
`public String getFilename( )`  
    - **Usage**  
 \* Returns the filename. Can be null.
- 
- *getProperties*  
`public Set getProperties( )`  
    - **Usage**  
 \* Returns the defined properties for this node.
    - **Returns** - a set of string
- 
- *getProperty*  
`public Object getProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns the value of a property
    - **Parameters**  
 \* **name** - the property name
    - **Returns** - null if the property was not previously set
- 
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns true if a property is defined for this node
    - **Parameters**  
 \* **name** - the name of the property
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**  
 \* **listener** - The PropertyChangeListener to be removed
-



- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* **propertyName** - The name of the property that was listened on.  
   \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
   \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
   \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
   \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
   \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
   \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

## 21.2.68 CLASS LongLiteral

---

This class represents the long literal nodes of the syntax tree

### DECLARATION

---

```
public class LongLiteral
extends koala.dynamicjava.tree.Literal
```

## CONSTRUCTORS

---

- *LongLiteral*

```
public LongLiteral( java.lang.String rep )
```

- **Usage**

- \* Initializes a literal

- **Parameters**

- \* **rep** - the representation of the literal

---

- *LongLiteral*

```
public LongLiteral( java.lang.String rep, java.lang.String fn, int bl, int  
bc, int el, int ec )
```

- **Usage**

- \* Initializes a literal

- **Parameters**

- \* **rep** - the representation of the literal

- \* **fn** - the filename

- \* **bl** - the begin line

- \* **bc** - the begin column

- \* **el** - the end line

- \* **ec** - the end column

## METHODS

---

- *parse*

```
private static Long parse( java.lang.String s )
```

- **Usage**

- \* Parse the representation of an integer

---

- *parseHexadecimal*

```
private static Long parseHexadecimal( java.lang.String s )
```

- **Usage**

- \* Parses an hexadecimal number

---

- *parseOctal*

```
private static Long parseOctal( java.lang.String s )
```

- **Usage**

- \* Parses an octal number

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Literal`

( in 21.2.67, page 1691)

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**  
 \* Allows a visitor to traverse the tree
  - **Parameters**  
 \* `visitor` - the visitor to accept
- *getRepresentation*  
`public String getRepresentation( )`
  - **Usage**  
 \* Returns the representation of this object
- *getType*  
`public Class getType( )`
  - **Usage**  
 \* Returns the type of this expression.  
 NOTE: the 'null' literal has a null type
- *getValue*  
`public Object getValue( )`
  - **Usage**  
 \* Returns the value of this expression
- *setRepresentation*  
`public void setRepresentation( java.lang.String s )`
  - **Usage**  
 \* Sets the representation of this object
  - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if `s` is null
- *setType*  
`public void setType( java.lang.Class c )`
  - **Usage**  
 \* Sets the type of this object
- *setValue*  
`public void setValue( java.lang.Object o )`
  - **Usage**  
 \* Sets the value of this object
  - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if `o` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String **name** )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**, java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int **i** )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int **i** )
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
 public void **setEndLine**( int **i** )
  - **Usage**
    - \* Sets the end line

---
- *setFilename*  
 public void **setFilename**( java.lang.String **s** )
  - **Usage**

- \* Sets the filename
- 
- *setProperty*  
`public void setProperty( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.69 CLASS LongType

---

This class represents the long type nodes of the syntax tree

#### DECLARATION

---

```
public class LongType
extends koala.dynamicjava.tree.PrimitiveType
```

#### CONSTRUCTORS

---

- *LongType*  
`public LongType( )`
    - **Usage**
      - \* Initializes the type
- 
- *LongType*  
`public LongType( java.lang.String fn, int bl, int bc, int el, int ec )`
    - **Usage**
      - \* Initializes the type
    - **Parameters**
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimitiveType

---

( in 21.2.89, page 1795)

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**
    - \* Allows a visitor to traverse the tree

- **Parameters**
  - \* visitor - the visitor to accept
- *getValue*


---

```
public Class getValue( )
```

  - **Usage**
    - \* Returns the value of this node
- *setValue*


---

```
public void setValue( java.lang.Class c )
```

  - **Usage**
    - \* Sets the value of this node
  - **Exceptions**
    - \* java.lang.IllegalArgumentException - if c is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Type

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* visitor - the visitor to accept
- *addPropertyChangeListener*


---

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* listener - The PropertyChangeListener to be added
- *addPropertyChangeListener*


---

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* propertyName - The name of the property to listen on.
    - \* listener - The PropertyChangeListener to be added

---



---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
-

- *getFilename*  
**public String getFilename( )**  
  - **Usage**  
 \* Returns the filename. Can be null.

---
- *getProperties*  
**public Set getProperties( )**  
  - **Usage**  
 \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns the value of a property
  - **Parameters**  
 \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns true if a property is defined for this node
  - **Parameters**  
 \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**  
 \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**  
  - **Usage**  
 \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**

- **Usage**
    - \* Sets the begin line
- *setEndColumn*


---

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*


---

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*


---

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*


---

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.70 CLASS MethodCall

---

This class represents the method call nodes of the syntax tree

#### DECLARATION

---

```
public abstract class MethodCall
extends koala.dynamicjava.tree.PrimaryExpression
implements ExpressionStatement
```

#### FIELDS

---

- public static final String METHOD\_NAME
  - The methodName property name
- public static final String ARGUMENTS
  - The arguments property name
- private String methodName
  - The method name
- private List arguments
  - The arguments

CONSTRUCTORS

---

• *MethodCall*

```
protected MethodCall( java.lang.String  mn, java.util.List  args,
    java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

– **Usage**

- \* Creates a new node

– **Parameters**

- \* **mn** - the field name
- \* **args** - the arguments. null if no arguments.
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

METHODS

---

• *getArguments*

```
public List getArguments( )
```

– **Usage**

- \* Returns the arguments.

- **Returns** - null if there is no argument

• *getMethodname*

```
public String getMethodName( )
```

– **Usage**

- \* Returns the method name

• *setArguments*

```
public void setArguments( java.util.List  l )
```

– **Usage**

- \* Sets the constructor arguments.

• *setMethodName*

```
public void setMethodName( java.lang.String  s )
```

– **Usage**

- \* Sets the method name

– **Exceptions**

- \* `java.lang.IllegalArgumentException` - if s is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*
    - public void **setProperty**( java.lang.String name, java.lang.Object value )
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.71 CLASS MethodDeclaration

---

This class represents method declarations in an AST

#### DECLARATION

---

```
public class MethodDeclaration
extends koala.dynamicjava.tree.Node
```

#### FIELDS

---

- public static final String ACCESS\_FLAGS
  - The accessFlags property name
- public static final String RETURN\_TYPE
  - The type property name
- public static final String NAME
  - The name property name
- public static final String PARAMETERS
  - The parameters property name
- public static final String EXCEPTIONS
  - The exceptions property name
- public static final String BODY
  - The body property name
- private int accessFlags
  - The access flags
- private Type returnType
  - The return type of this method
- private String name
  - The name of this method



- private List parameters
  - The parameters
- private List exceptions
  - The exceptions
- private BlockStatement body
  - The body of the method

## CONSTRUCTORS

---

- *MethodDeclaration*

```
public MethodDeclaration( int  flags, koala.dynamicjava.tree.Type  type,
java.lang.String  name, java.util.List  params, java.util.List  excepts,
koala.dynamicjava.tree.BlockStatement  body )
```

- **Usage**
    - \* Creates a new method declaration
  - **Parameters**
    - \* **flags** - the access flags
    - \* **type** - the return type of this method
    - \* **name** - the name of the method to declare
    - \* **params** - the parameters list
    - \* **excepts** - the exception list
    - \* **body** - the body statement
- 

- *MethodDeclaration*

```
public MethodDeclaration( int  flags, koala.dynamicjava.tree.Type  type,
java.lang.String  name, java.util.List  params, java.util.List  excepts,
koala.dynamicjava.tree.BlockStatement  body, java.lang.String  fn, int  bl,
int  bc, int  el, int  ec )
```

- **Usage**
  - \* Creates a new method declaration
- **Parameters**
  - \* **flags** - the access flags
  - \* **type** - the return type of this method
  - \* **name** - the name of the method to declare
  - \* **params** - the parameters list
  - \* **excepts** - the exception list
  - \* **body** - the body statement
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getAccessFlags*

```
public int getAccessFlags( )
```

– **Usage**

\* Returns the access flags for this method

---

• *getBody*

```
public BlockStatement getBody( )
```

– **Usage**

\* Returns the body of the method, null if the method is abstract

---

• *getExceptions*

```
public List getExceptions( )
```

– **Usage**

\* Returns the list of the exception thrown by this method

– **Returns** - a list of string

---

• *getName*

```
public String getName( )
```

– **Usage**

\* Returns the name of this method

---

• *getParameters*

```
public List getParameters( )
```

– **Usage**

\* Returns the parameters list

---

• *getReturnType*

```
public Type getReturnType( )
```

– **Usage**

\* Gets the return type of this method

---

• *setAccessFlags*

```
public void setAccessFlags( int  f )
```

– **Usage**

- \* Sets the access flags for this constructor

---

- *setBody*  
`public void setBody( koala.dynamicjava.tree.BlockStatement bs )`
    - **Usage**  
 \* Sets the body

---

- *setExceptions*  
`public void setExceptions( java.util.List l )`
    - **Usage**  
 \* Sets the exceptions list
    - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if `l` is null

---

- *setName*  
`public void setName( java.lang.String s )`
    - **Usage**  
 \* Sets the method's name
    - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if `s` is null

---

- *setParameters*  
`public void setParameters( java.util.List l )`
    - **Usage**  
 \* Sets the parameters list
    - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if `l` is null

---

- *setReturnType*  
`public void setReturnType( koala.dynamicjava.tree.Type t )`
    - **Usage**  
 \* Sets the return type of this method
    - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if `t` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
      - **Usage**  
 \* Allows a visitor to traverse the tree
      - **Parameters**  
 \* `visitor` - the visitor to accept
-

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.
    - The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added
- 

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
-

- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line
 

---
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code
 

---
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line
 

---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code
 

---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.
 

---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string
 

---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set
 

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property
 

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
 

---

- **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.72 CLASS MinusExpression

---

This class represents the minus expression nodes of the syntax tree

DECLARATION

---

```
public class MinusExpression
extends koala.dynamicjava.tree.UnaryExpression
```

CONSTRUCTORS

---

• *MinusExpression*

```
public MinusExpression( koala.dynamicjava.tree.Expression  exp )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **exp** - the target expression

• *MinusExpression*

```
public MinusExpression( koala.dynamicjava.tree.Expression  exp,
java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **exp** - the target expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.UnaryExpression

---

( in 21.2.120, page 1938)

• *getExpression*

```
public Expression getExpression( )
```

– **Usage**

\* Returns the target expression

- 
- *setExpression*  
 public void **setExpression**( koala.dynamicjava.tree.Expression e )  
 – **Usage**  
   \* Sets the target expression  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if e is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* visitor - the visitor to accept
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener to the listener list.  
   The listener is registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be added
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.  
 – **Parameters**  
   \* propertyName - The name of the property to listen on.  
   \* listener - The PropertyChangeListener to be added
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String propertyName, boolean oldValue, boolean newValue )  
 – **Usage**  
   \* Report a bound property update to any registered listeners.  
   No event is fired if old and new are equal and non-null.  
 – **Parameters**



- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *getBeginColumn*

public int **getBeginColumn**( )

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

public int **getBeginLine**( )

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

public int **getEndColumn**( )

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

public int **getEndLine**( )

- **Usage**

- \* Returns the end line of this node in the source code

---

- *getFilename*

public String **getFilename**( )

- **Usage**

- \* Returns the filename. Can be null.

---

- *getProperties*

public Set **getProperties**( )

- **Usage**

- \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*
  - public Object **getProperty**( java.lang.String name )
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* name - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*
  - public boolean **hasProperty**( java.lang.String name )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* name - the name of the property

---
- *removePropertyChangeListener*
  - public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* listener - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*
  - public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* propertyName - The name of the property that was listened on.
    - \* listener - The PropertyChangeListener to be removed

---
- *setBeginColumn*
  - public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*
  - public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*
  - public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.73 CLASS MultiplyAssignExpression

---

This class represents the multiply assign expression nodes of the syntax tree

#### DECLARATION

---

```
public class MultiplyAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

#### CONSTRUCTORS

---

- *MultiplyAssignExpression*  
 public **MultiplyAssignExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**  
   \* **lexp** - the LHS expression  
   \* **rexp** - the RHS expression  


---
- *MultiplyAssignExpression*  
 public **MultiplyAssignExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int  
 bc, int el, int ec )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---

( in 21.2.10, page 1417)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**

- \* Returns the left hand side expression

- *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**

- \* Returns the right hand side expression

- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression  exp )
```

- **Usage**

- \* Sets the left hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null

- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression  exp )
```

- **Usage**

- \* Sets the right hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

\* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

- **Usage**

- \* Sets the value of a property

- **Parameters**

- \* **name** - the property name

- \* **value** - the new value to set

## 21.2.74 CLASS MultiplyExpression

---

This class represents the multiply expression nodes of the syntax tree

### DECLARATION

---

```
public class MultiplyExpression
extends koala.dynamicjava.tree.BinaryExpression
```

### CONSTRUCTORS

---

- *MultiplyExpression*

```
public MultiplyExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression

- \* **rexp** - the RHS expression

---

- *MultiplyExpression*

```
public MultiplyExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression

- \* **rexp** - the RHS expression

- \* **fn** - the filename

- \* **bl** - the begin line

- \* **bc** - the begin column

- \* **el** - the end line

- \* **ec** - the end column



METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

– **Usage**

\* Returns the left hand side expression

• *getRightExpression*

```
public Expression getRightExpression( )
```

– **Usage**

\* Returns the right hand side expression

• *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression  exp )
```

– **Usage**

\* Sets the left hand side expression

– **Exceptions**

\* java.lang.IllegalArgumentException - if exp is null

• *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression  exp )
```

– **Usage**

\* Sets the right hand side expression

– **Exceptions**

\* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.75 CLASS Node

---

This class represents the nodes of the syntax tree

#### DECLARATION

---

```
public abstract class Node
extends java.lang.Object
```

#### FIELDS

---

- public static final String FILENAME
  - The filename property name
- public static final String BEGIN\_LINE
  - The beginLine property name
- public static final String END\_LINE
  - The endLine property name
- public static final String BEGIN\_COLUMN
  - The beginColumn property name
- public static final String END\_COLUMN
  - The endColumn property name
- private String filename
  - The filename
- private int beginLine
  - The begin line in the source code
- private int beginColumn
  - The begin column in the begin line
- private int endLine
  - The end line in the source code
- private int endColumn
  - The end column in the end line
- private PropertyChangeSupport propertyChangeSupport
  - The support for the property change mechanism
- private Map properties
  - The properties

CONSTRUCTORS

---

- *Node*

```
protected Node( java.lang.String fn, int bl, int bc, int el, int ec )
```

- **Usage**

- \* Initializes the node

- **Parameters**

- \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS

---

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

---

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
  - \* **oldValue** - The old value of the property.
  - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
  - \* **oldValue** - The old value of the property.
  - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
  - \* **oldValue** - The old value of the property.
  - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

public int **getBeginColumn**( )

– **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

public int **getBeginLine**( )

– **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

public int **getEndColumn**( )

– **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

public int **getEndLine**( )

- **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*

```
public String getFilename( )
```

    - **Usage**
      - \* Returns the filename. Can be null.

---
- *getProperties*

```
public Set getProperties( )
```

    - **Usage**
      - \* Returns the defined properties for this node.
    - **Returns** - a set of string

---
- *getProperty*

```
public Object getProperty( java.lang.String name )
```

    - **Usage**
      - \* Returns the value of a property
    - **Parameters**
      - \* **name** - the property name
    - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener(
java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String
propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**



- \* **propertyName** - The name of the property that was listened on.
- \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

**public void setBeginColumn( int i )**

– **Usage**

- \* Sets the begin column
- 

- *setBeginLine*

**public void setBeginLine( int i )**

– **Usage**

- \* Sets the begin line
- 

- *setEndColumn*

**public void setEndColumn( int i )**

– **Usage**

- \* Sets the end column
- 

- *setEndLine*

**public void setEndLine( int i )**

– **Usage**

- \* Sets the end line
- 

- *setFilename*

**public void setFilename( java.lang.String s )**

– **Usage**

- \* Sets the filename
- 

- *setProperty*

**public void setProperty( java.lang.String name, java.lang.Object value )**

– **Usage**

- \* Sets the value of a property

– **Parameters**

- \* **name** - the property name
- \* **value** - the new value to set

## 21.2.76 CLASS NotEqualExpression

---

This class represents the not equal expression nodes of the syntax tree

### DECLARATION

---

```
public class NotEqualExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *NotEqualExpression*

```
public NotEqualExpression( koala.dynamicjava.tree.Expression lexp,
                           koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression

---

• *NotEqualExpression*

```
public NotEqualExpression( koala.dynamicjava.tree.Expression lexp,
                           koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
                           bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

## – Usage

\* Returns the left hand side expression

---

• *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.

\* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code

- 
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
   \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
   \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns the value of a property  
 – **Parameters**  
   \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns true if a property is defined for this node  
 – **Parameters**  
   \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener from the listener list.  
   This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* propertyName - The name of the property that was listened on.  
   \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
   \* Sets the begin column

---

  - *setBeginLine*  
 public void **setBeginLine**( int i )

- **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.77 CLASS NotExpression

---

This class represents the not expression nodes of the syntax tree

### DECLARATION

---

```
public class NotExpression
extends koala.dynamicjava.tree.UnaryExpression
```

### CONSTRUCTORS

---

- *NotExpression*

```
public NotExpression( koala.dynamicjava.tree.Expression exp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the target expression

---

- *NotExpression*

```
public NotExpression( koala.dynamicjava.tree.Expression exp,
java.lang.String fn, int bl, int bc, int el, int ec )
```

- **Usage**
  - \* Initializes the expression
- **Parameters**
  - \* **exp** - the target expression
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`

- **Usage**
  - \* Allows a visitor to traverse the tree
- **Parameters**
  - \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.UnaryExpression

---

( in 21.2.120, page 1938)

- *getExpression*  
`public Expression getExpression( )`
    - **Usage**
      - \* Returns the target expression
- 
- *setExpression*  
`public void setExpression( koala.dynamicjava.tree.Expression e )`
    - **Usage**
      - \* Sets the target expression
    - **Exceptions**
      - \* `java.lang.IllegalArgumentException` - if e is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**

- \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept
- - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be added
- - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added
- - *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.
- - *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.
- - *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.



- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

- 
- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* `name` - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* `name` - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**

- \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
- **Parameters**
  - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*  

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*  

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.78 CLASS NullLiteral

---

This class represents the null literal nodes of the syntax tree

DECLARATION

---

```
public class NullLiteral
extends koala.dynamicjava.tree.Literal
```

CONSTRUCTORS

---

- *NullLiteral*  
 public **NullLiteral**( )  
 – **Usage**  
   \* Initializes a literal

---

- *NullLiteral*  
 public **NullLiteral**( java.lang.String fn, int bl, int bc, int el, int ec )  
 – **Usage**  
   \* Initializes a literal  
 – **Parameters**  
   \* **fn** - the filename  
   \* **bl** - the begin line  
   \* **bc** - the begin column  
   \* **el** - the end line  
   \* **ec** - the end column

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Literal

---

( in 21.2.67, page 1691)

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* **visitor** - the visitor to accept

---

- *getRepresentation*  
 public String **getRepresentation**( )  
 – **Usage**  
   \* Returns the representation of this object

---

- *getType*  
 public Class **getType**( )  
 – **Usage**  
   \* Returns the type of this expression.  
   NOTE: the 'null' literal has a null type

---

- *getValue*  
 public Object **getValue**( )

- **Usage**
    - \* Returns the value of this expression
- 
- *setRepresentation*

```
public void setRepresentation( java.lang.String s )
```

    - **Usage**
      - \* Sets the representation of this object
    - **Exceptions**
      - \* `java.lang.IllegalArgumentException` - if s is null
- 
- *setType*

```
public void setType( java.lang.Class c )
```

    - **Usage**
      - \* Sets the type of this object
- 
- *setValue*

```
public void setValue( java.lang.Object o )
```

    - **Usage**
      - \* Sets the value of this object
    - **Exceptions**
      - \* `java.lang.IllegalArgumentException` - if o is null

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* `visitor` - the visitor to accept
- 
- *addChangeListener*

```
public void addChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

- 
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.

- **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.79 CLASS ObjectFieldAccess

---

This class represents the field access nodes of the syntax tree

#### DECLARATION

---

```
public class ObjectFieldAccess
extends koala.dynamicjava.tree.FieldAccess
implements ExpressionContainer
```

FIELDS

---

- private Expression expression
  - The expression on which this field access applies

CONSTRUCTORS

---

- *ObjectFieldAccess*  
 public ObjectFieldAccess( koala.dynamicjava.tree.Expression exp,  
 java.lang.String fln )
    - **Usage**
      - \* Creates a new field access node
    - **Parameters**
      - \* **exp** - the expression on which this field access applies
      - \* **fln** - the field name
- 
- *ObjectFieldAccess*  
 public ObjectFieldAccess( koala.dynamicjava.tree.Expression exp,  
 java.lang.String fln, java.lang.String fn, int bl, int bc, int el, int  
 ec )
    - **Usage**
      - \* Creates a new field access node
    - **Parameters**
      - \* **exp** - the expression on which this field access applies
      - \* **fln** - the field name
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS

---

- *acceptVisitor*  
 public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* **visitor** - the visitor to accept
- 
- *getExpression*  
 public Expression getExpression( )
    - **Usage**



\* Returns the expression on which this field access applies

---

- *setExpression*

**public void setExpression( koala.dynamicjava.tree.Expression e )**

- **Usage**

\* Sets the expression on which this field access applies

- **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.FieldAccess`

( in 21.2.43, page 1575)

- *getFieldName*

**public String getFieldName( )**

- **Usage**

\* Returns the field name

---

- *setFieldName*

**public void setFieldName( java.lang.String s )**

- **Usage**

\* Sets the field name

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

( in 21.2.88, page 1792)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

- *acceptVisitor*

**public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**

- **Usage**

\* Allows a visitor to traverse the tree

- **Parameters**

\* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*

**public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )**

- **Usage**

\* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.

- **Parameters**
    - \* *listener* - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* *propertyName* - The name of the property to listen on.
    - \* *listener* - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*

```
public String getFilename( )
```

  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*

```
public Set getProperties( )
```

  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**

- \* Removes a PropertyChangeListener for a specific property.
- **Parameters**
  - \* **propertyName** - The name of the property that was listened on.
  - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column
- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line
- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column
- *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line
- *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.80 CLASS ObjectMethodCall

---

This class represents the method call nodes of the syntax tree

### DECLARATION

---

```
public class ObjectMethodCall
extends koala.dynamicjava.tree.MethodCall
implements ExpressionContainer
```

FIELDS

---

- private Expression expression
  - The expression on which this method call applies

CONSTRUCTORS

---

- *ObjectMethodCall*  
 public **ObjectMethodCall**( koala.dynamicjava.tree.Expression exp,  
 java.lang.String mn, java.util.List args, java.lang.String fn, int bl,  
 int bc, int el, int ec )
  - **Usage**
    - \* Creates a new node
  - **Parameters**
    - \* **exp** - the expression on which this method call applies
    - \* **mn** - the field name
    - \* **args** - the arguments. Can be null.
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept
- *getExpression*  
 public Expression **getExpression**( )
  - **Usage**
    - \* Returns the expression on which this method call applies
- *setExpression*  
 public void **setExpression**( koala.dynamicjava.tree.Expression e )
  - **Usage**
    - \* Sets the expression on which this method call applies

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.MethodCall`

---

( in 21.2.70, page 1706)

- *getArguments*  
`public List getArguments( )`
  - **Usage**  
 \* Returns the arguments.
  - **Returns** - null if there is no argument
- *getMethodName*  
`public String getMethodName( )`
  - **Usage**  
 \* Returns the method name
- *setArguments*  
`public void setArguments( java.util.List l )`
  - **Usage**  
 \* Sets the constructor arguments.
- *setMethodName*  
`public void setMethodName( java.lang.String s )`
  - **Usage**  
 \* Sets the method name
  - **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if s is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**  
 \* Allows a visitor to traverse the tree
  - **Parameters**  
 \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`

- **Usage**
  - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
- **Parameters**
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*  
**public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

- 
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---



- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
 \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
 \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
 \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
 \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
 \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
 \* Sets the value of a property  
 – **Parameters**  
 \* **name** - the property name  
 \* **value** - the new value to set

### 21.2.81 CLASS OrExpression

---

This class represents the or expression nodes of the syntax tree

#### DECLARATION

---

```
public class OrExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *OrExpression*

```
public OrExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression

---

• *OrExpression*

```
public OrExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

## – Usage

\* Returns the left hand side expression

---

• *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.

---

\* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code

- 
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
 \* Sets the begin column

---

  - *setBeginLine*  
 public void **setBeginLine**( int i )

- **Usage**
    - \* Sets the begin line
- - *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
- - *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- - *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- - *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.82 CLASS PackageDeclaration

---

This class represents the package declarations

#### DECLARATION

---

```
public class PackageDeclaration
extends koala.dynamicjava.tree.Node
```

#### FIELDS

---

- public static final String NAME
  - The name property name
- private String name
  - The name of the package

CONSTRUCTORS

---

• *PackageDeclaration*

```
public PackageDeclaration( java.util.List ident )
```

– **Usage**

\* Creates a new package declaration node

– **Parameters**

\* **ident** - a list of tokens that represents a package name.  
The list can be null.

---

• *PackageDeclaration*

```
public PackageDeclaration( java.util.List ident, java.lang.String fn, int
bl, int bc, int el, int ec )
```

– **Usage**

\* Creates a new package declaration node

– **Parameters**

\* **ident** - a list of tokens that represents a package name.  
The list can be null.  
\* **fn** - the filename  
\* **bl** - the begin line  
\* **bc** - the begin column  
\* **el** - the end line  
\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getName*

```
public String getName( )
```

– **Usage**

\* Returns the name of the imported class or package

---

• *setName*

```
public void setName( java.lang.String s )
```

– **Usage**

\* Sets the name

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if `s` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.



- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.83 CLASS **PlusExpression**

---

This class represents the plus expression nodes of the syntax tree

#### DECLARATION

---

```
public class PlusExpression
extends koala.dynamicjava.tree.UnaryExpression
```

#### CONSTRUCTORS

---

- *PlusExpression*  

```
public PlusExpression( koala.dynamicjava.tree.Expression  exp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the target expression

---
- *PlusExpression*  

```
public PlusExpression( koala.dynamicjava.tree.Expression  exp,
java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the target expression
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

#### METHODS

---

- *acceptVisitor*  

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.UnaryExpression`

---

( in 21.2.120, page 1938)

- *getExpression*  
`public Expression getExpression( )`  
 – **Usage**  
 \* Returns the target expression

---

- *setExpression*  
`public void setExpression( koala.dynamicjava.tree.Expression e )`  
 – **Usage**  
 \* Sets the target expression  
 – **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if e is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
 – **Usage**  
 \* Allows a visitor to traverse the tree  
 – **Parameters**  
 \* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`  
 – **Usage**  
 \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.  
 – **Parameters**  
 \* `listener` - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`  
 – **Usage**  
 \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.  
 – **Parameters**  
 \* `propertyName` - The name of the property to listen on.  
 \* `listener` - The PropertyChangeListener to be added

- 
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, boolean **oldValue**, boolean **newValue** )  
    - **Usage**
      - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
    - **Parameters**
      - \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )  
    - **Usage**
      - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
    - **Parameters**
      - \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )  
    - **Usage**
      - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
    - **Parameters**
      - \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 
- *getBeginColumn*  
public int **getBeginColumn**( )  
    - **Usage**
      - \* Returns the begin column of this node in the begin line
- 
- *getBeginLine*  
public int **getBeginLine**( )  
    - **Usage**
      - \* Returns the begin line of this node in the source code
- 
- *getEndColumn*  
public int **getEndColumn**( )  
    - **Usage**
      - \* Returns the end column of this node in the end line
- 
- *getEndLine*  
public int **getEndLine**( )  
    - **Usage**
      - \* Returns the end line of this node in the source code
-

- *getFilename*  
**public String getFilename( )**  
  - **Usage**  
 \* Returns the filename. Can be null.

---
- *getProperties*  
**public Set getProperties( )**  
  - **Usage**  
 \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns the value of a property
  - **Parameters**  
 \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns true if a property is defined for this node
  - **Parameters**  
 \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**  
 \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**  
  - **Usage**  
 \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**

- **Usage**
    - \* Sets the begin line
- - *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
- - *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- - *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- - *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.84 CLASS PostDecrement

---

This class represents the post decrement nodes of the syntax tree

#### DECLARATION

---

```
public class PostDecrement
extends koala.dynamicjava.tree.UnaryExpression
implements ExpressionStatement
```

#### CONSTRUCTORS

---

- *PostDecrement*

```
public PostDecrement( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **exp** - the post decremented expression
- - *PostDecrement*

```
public PostDecrement( koala.dynamicjava.tree.Expression exp,
java.lang.String fn, int bl, int bc, int el, int ec )
```

- **Usage**
  - \* Initializes the expression
- **Parameters**
  - \* **exp** - the post decremented expression
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*  
`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.UnaryExpression

---

( in 21.2.120, page 1938)

- *getExpression*  
`public Expression getExpression( )`
  - **Usage**
    - \* Returns the target expression
- *setExpression*  
`public void setExpression( koala.dynamicjava.tree.Expression e )`
  - **Usage**
    - \* Sets the target expression
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if e is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**



- \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept
- - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be added
- - *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added
- - *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.
- - *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.
- - *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.

- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

- 
- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* `name` - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* `name` - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**

- \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
- **Parameters**
  - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*  

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*  

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.85 CLASS PostIncrement

---

This class represents the post increment nodes of the syntax tree

DECLARATION

---

```
public class PostIncrement
extends koala.dynamicjava.tree.UnaryExpression
implements ExpressionStatement
```

CONSTRUCTORS

---

- *PostIncrement*  

```
public PostIncrement( koala.dynamicjava.tree.Expression  exp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the post incremented expression

---
- *PostIncrement*  

```
public PostIncrement( koala.dynamicjava.tree.Expression  exp,
java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **exp** - the post incremented expression
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS

---

- *acceptVisitor*  

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.UnaryExpression

---

( in 21.2.120, page 1938)

- *getExpression*  

```
public Expression getExpression( )
```

  - **Usage**

- \* Returns the target expression
- 
- *setExpression*  
 public void **setExpression**( koala.dynamicjava.tree.Expression e )
    - **Usage**
      - \* Sets the target expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if e is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.
      - \* listener - The PropertyChangeListener to be added
- 
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String propertyName, boolean oldValue, boolean newValue )
    - **Usage**
      - \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.

- **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  

```
public int getBeginLine( )
```

  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  

```
public int getEndColumn( )
```

  - **Usage**
    - \* Returns the end column of this node in the end line

---
- *getEndLine*  

```
public int getEndLine( )
```

  - **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  

```
public String getFilename( )
```

  - **Usage**
    - \* Returns the filename. Can be null.

---
- *getProperties*  

```
public Set getProperties( )
```

- **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* name - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* name - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* listener - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* propertyName - The name of the property that was listened on.
    - \* listener - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

## 21.2.86 CLASS PreDecrement

---

This class represents the pre decrement nodes of the syntax tree

### DECLARATION

---

```
public class PreDecrement
extends koala.dynamicjava.tree.UnaryExpression
implements ExpressionStatement
```

### CONSTRUCTORS

---

- *PreDecrement*  
 public **PreDecrement**( koala.dynamicjava.tree.Expression exp )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**  
   \* **exp** - the pre decremented expression

---

- *PreDecrement*  
 public **PreDecrement**( koala.dynamicjava.tree.Expression exp,  
   java.lang.String fn, int bl, int bc, int el, int ec )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**  
   \* **exp** - the pre decremented expression  
   \* **fn** - the filename  
   \* **bl** - the begin line  
   \* **bc** - the begin column  
   \* **el** - the end line  
   \* **ec** - the end column



METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.UnaryExpression

---

( in 21.2.120, page 1938)

• *getExpression*

```
public Expression getExpression( )
```

– **Usage**

\* Returns the target expression

• *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression  e )
```

– **Usage**

\* Sets the target expression

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener )
```

– **Usage**

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

- **Parameters**
    - \* *listener* - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* *propertyName* - The name of the property to listen on.
    - \* *listener* - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*

```
public String getFilename( )
```

  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*

```
public Set getProperties( )
```

  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**

- \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- 
- *setBeginColumn*  
 public void **setBeginColumn**( int i )
    - **Usage**
      - \* Sets the begin column
- 
- *setBeginLine*  
 public void **setBeginLine**( int i )
    - **Usage**
      - \* Sets the begin line
- 
- *setEndColumn*  
 public void **setEndColumn**( int i )
    - **Usage**
      - \* Sets the end column
- 
- *setEndLine*  
 public void **setEndLine**( int i )
    - **Usage**
      - \* Sets the end line
- 
- *setFilename*  
 public void **setFilename**( java.lang.String s )
    - **Usage**
      - \* Sets the filename
- 
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.87 CLASS PreIncrement

---

This class represents the pre increment nodes of the syntax tree

#### DECLARATION

---

```
public class PreIncrement
extends koala.dynamicjava.tree.UnaryExpression
implements ExpressionStatement
```

CONSTRUCTORS

---

• *PreIncrement*

```
public PreIncrement( koala.dynamicjava.tree.Expression exp )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **exp** - the pre incremented expression

---

• *PreIncrement*

```
public PreIncrement( koala.dynamicjava.tree.Expression exp,  
java.lang.String fn, int bl, int bc, int el, int ec )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **exp** - the pre incremented expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.UnaryExpression

---

( in 21.2.120, page 1938)

• *getExpression*

```
public Expression getExpression( )
```

– **Usage**

\* Returns the target expression

---

• *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression e )
```

– **Usage**

\* Sets the target expression

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String **name** )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**, java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int **i** )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int **i** )
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
 public void **setEndLine**( int **i** )
  - **Usage**
    - \* Sets the end line

---
- *setFilename*  
 public void **setFilename**( java.lang.String **s** )
  - **Usage**



- \* Sets the filename
- 
- *setProperty*  

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

## 21.2.88 CLASS PrimaryExpression

---

This class represents the primary expression nodes of the syntax tree

### DECLARATION

---

```
public abstract class PrimaryExpression
extends koala.dynamicjava.tree.Expression
```

### CONSTRUCTORS

---

- *PrimaryExpression*  

```
protected PrimaryExpression( java.lang.String fn, int bl, int bc, int
el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree

- **Parameters**
    - \* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener  
will be invoked only when a call on firePropertyChange names that  
specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue,
int newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName,
java.lang.Object oldValue, java.lang.Object newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.

\* **newValue** - The new value of the property.

---

- *getBeginColumn*

public int **getBeginColumn**( )

– **Usage**

\* Returns the begin column of this node in the begin line

---

- *getBeginLine*

public int **getBeginLine**( )

– **Usage**

\* Returns the begin line of this node in the source code

---

- *getEndColumn*

public int **getEndColumn**( )

– **Usage**

\* Returns the end column of this node in the end line

---

- *getEndLine*

public int **getEndLine**( )

– **Usage**

\* Returns the end line of this node in the source code

---

- *getFilename*

public String **getFilename**( )

– **Usage**

\* Returns the filename. Can be null.

---

- *getProperties*

public Set **getProperties**( )

– **Usage**

\* Returns the defined properties for this node.

– **Returns** - a set of string

---

- *getProperty*

public Object **getProperty**( java.lang.String **name** )

– **Usage**

\* Returns the value of a property

– **Parameters**

\* **name** - the property name

– **Returns** - null if the property was not previously set

---

- *hasProperty*

public boolean **hasProperty**( java.lang.String **name** )

– **Usage**

\* Returns true if a property is defined for this node

– **Parameters**

\* **name** - the name of the property

---

- *removePropertyChangeListener*

public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )

– **Usage**

- \* Removes a `PropertyChangeListener` from the listener list.  
This removes a `PropertyChangeListener` that was registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a `PropertyChangeListener` for a specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property that was listened on.
    - \* `listener` - The `PropertyChangeListener` to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* `name` - the property name
    - \* `value` - the new value to set

### 21.2.89 CLASS `PrimitiveType`

---

This class represents the primitive type nodes of the syntax tree

DECLARATION

---

```
public abstract class PrimitiveType
extends koala.dynamicjava.tree.Type
```

FIELDS

---

- public static final String VALUE
  - The value property name
- private Class value
  - The value of the node

CONSTRUCTORS

---

- *PrimitiveType*  
 protected **PrimitiveType**( java.lang.Class val, java.lang.String fn, int bl, int bc, int el, int ec )
  - **Usage**
    - \* Initializes the type
  - **Parameters**
    - \* val - the value of this type
    - \* fn - the filename
    - \* bl - the begin line
    - \* bc - the begin column
    - \* el - the end line
    - \* ec - the end column

METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *getValue*  
 public Class **getValue**( )
    - **Usage**
      - \* Returns the value of this node
-

- *setValue*  
`public void setValue( java.lang.Class c )`
  - **Usage**
    - \* Sets the value of this node
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if c is null

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`
  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added
- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *getBeginColumn*

public int **getBeginColumn**( )

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

public int **getBeginLine**( )

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

public int **getEndColumn**( )

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

public int **getEndLine**( )

- **Usage**

- \* Returns the end line of this node in the source code

---

- *getFilename*

public String **getFilename**( )

- **Usage**

- \* Returns the filename. Can be null.

---

- *getProperties*

public Set **getProperties**( )

- **Usage**

- \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*
  - public Object **getProperty**( java.lang.String name )
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* name - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*
  - public boolean **hasProperty**( java.lang.String name )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* name - the name of the property

---
- *removePropertyChangeListener*
  - public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* listener - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*
  - public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* propertyName - The name of the property that was listened on.
    - \* listener - The PropertyChangeListener to be removed

---
- *setBeginColumn*
  - public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*
  - public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*
  - public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---



- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

## 21.2.90 CLASS QualifiedName

---

This class represents the qualified name nodes of the syntax tree

### DECLARATION

---

```
public class QualifiedName
extends koala.dynamicjava.tree.PrimaryExpression
implements LeftHandSide
```

### FIELDS

---

- public static final String IDENTIFIERS  
 – The identifiers property name
- public static final String REPRESENTATION  
 – The representation property name
- private List identifiers  
 – The identifiers (tokens) that compose this name
- private String representation  
 – The representation of this object

CONSTRUCTORS

---

• *QualifiedName*

```
public QualifiedName( java.util.List ids )
```

## – Usage

\* Creates a new qualified name

## – Parameters

\* **ids** - the identifiers (IdentifierTokens) that compose this name

---

• *QualifiedName*

```
public QualifiedName( java.util.List ids, java.lang.String fn, int bl,
int bc, int el, int ec )
```

## – Usage

\* Creates a new qualified name

## – Parameters

\* **ids** - the identifiers (IdentifierTokens) that compose this name

\* **fn** - the filename

\* **bl** - the begin line

\* **bc** - the begin column

\* **el** - the end line

\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

---

• *getIdentifiers*

```
public List getIdentifiers( )
```

## – Usage

\* Returns the identifiers that compose this name

---

• *getRepresentation*

```
public String getRepresentation( )
```

## – Usage

\* Returns the representation of this object

---

• *setIdentifier*

```
public void setIdentifier( java.util.List l )
```

- **Usage**
  - \* Sets the identifiers that compose this name. Update representation
- **Exceptions**
  - \* `java.lang.IllegalArgumentException` - if `l` is null

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.
-

- *getProperties*  
**public Set getProperties( )**
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
**public void setEndColumn( int i )**

- **Usage**
    - \* Sets the end column
- 
- *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- 
- *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.91 CLASS ReferenceType

This class represents the reference type nodes of the syntax tree

#### DECLARATION

```
public class ReferenceType
extends koala.dynamicjava.tree.Type
```

#### FIELDS

- public static final String REPRESENTATION
  - The representation property name
- private String representation
  - The representation of this type

#### CONSTRUCTORS

- *ReferenceType*

```
public ReferenceType( java.util.List ids )
```

  - **Usage**
    - \* Initializes the type
  - **Parameters**

\* **ids** - the list of the tokens that compose the type name

---

- *ReferenceType*

```
public ReferenceType( java.util.List  ids, java.lang.String  fn, int  bl,
int  bc, int  el, int  ec )
```

- **Usage**

- \* Initializes the type

- **Parameters**

- \* **ids** - the list of the tokens that compose the type name
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

---

- *ReferenceType*

```
public ReferenceType( java.lang.String  rep )
```

- **Usage**

- \* Initializes the type

- **Parameters**

- \* **rep** - the type name

---

- *ReferenceType*

```
public ReferenceType( java.lang.String  rep, java.lang.String  fn, int  bl,
int  bc, int  el, int  ec )
```

- **Usage**

- \* Initializes the type

- **Parameters**

- \* **rep** - the type name
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getRepresentation*

```
public String getRepresentation( )
```

- **Usage**

- \* Returns the representation of this type

---

- *setRepresentation*

```
public void setRepresentation( java.lang.String s )
```

- **Usage**

- \* Sets the representation of this type

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if s is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

- **Parameters**

- \* `propertyName` - The name of the property to listen on.
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

- **Usage**



- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.
-

- *getProperties*  
**public Set getProperties( )**
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
**public void setEndColumn( int i )**

- **Usage**
    - \* Sets the end column
- 
- *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- 
- *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.92 CLASS RemainderAssignExpression

This class represents the remainder assign expression nodes of the syntax tree

#### DECLARATION

```
public class RemainderAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

#### CONSTRUCTORS

- *RemainderAssignExpression*

```
public RemainderAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **lexp** - the LHS expression
      - \* **rexp** - the RHS expression
- 
- *RemainderAssignExpression*

```
public RemainderAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

    - **Usage**

- \* Initializes the expression
- **Parameters**
  - \* **lexp** - the LHS expression
  - \* **rexp** - the RHS expression
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor **visitor** )
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---

( in 21.2.10, page 1417)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*  
 public Expression **getLeftExpression**( )
  - **Usage**
    - \* Returns the left hand side expression
- *getRightExpression*  
 public Expression **getRightExpression**( )
  - **Usage**
    - \* Returns the right hand side expression
- *setLeftExpression*  
 public void **setLeftExpression**( koala.dynamicjava.tree.Expression **exp** )
  - **Usage**
    - \* Sets the left hand side expression
  - **Exceptions**
    - \* java.lang.IllegalArgumentException - if exp is null
- *setRightExpression*  
 public void **setRightExpression**( koala.dynamicjava.tree.Expression **exp** )
  - **Usage**
    - \* Sets the right hand side expression
  - **Exceptions**
    - \* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  

```
public int getBeginLine( )
```

  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  

```
public int getEndColumn( )
```

  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*  

```
public int getEndLine( )
```

  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*  

```
public String getFilename( )
```

  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*  

```
public Set getProperties( )
```

  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---

- *getProperty*  

```
public Object getProperty( java.lang.String  name )
```

  - **Usage**
    - \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.93 CLASS RemainderExpression

---

This class represents the remainder expression nodes of the syntax tree

#### DECLARATION

---

```
public class RemainderExpression
extends koala.dynamicjava.tree.BinaryExpression
```

#### CONSTRUCTORS

---

- *RemainderExpression*

```
public RemainderExpression( koala.dynamicjava.tree.Expression  lexp,
koala.dynamicjava.tree.Expression  rexp )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **lexp** - the LHS expression
      - \* **rexp** - the RHS expression
- 
- *RemainderExpression*

```
public RemainderExpression( koala.dynamicjava.tree.Expression  lexp,
koala.dynamicjava.tree.Expression  rexp, java.lang.String  fn, int  bl, int
bc, int  el, int  ec )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **lexp** - the LHS expression
      - \* **rexp** - the RHS expression
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column



## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**

- \* Returns the left hand side expression

---

- *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**

- \* Returns the right hand side expression

---

- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression  exp )
```

- **Usage**

- \* Sets the left hand side expression

- **Exceptions**

- \* java.lang.IllegalArgumentException - if exp is null

---

- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression  exp )
```

- **Usage**

- \* Sets the right hand side expression

- **Exceptions**

- \* java.lang.IllegalArgumentException - if exp is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* **listener** - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

## – Parameters

\* **propertyName** - The name of the property to listen on.  
\* **listener** - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* **propertyName** - The programmatic name of the property that was changed.  
\* **oldValue** - The old value of the property.  
\* **newValue** - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
    java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.94 CLASS ReturnStatement

---

This class represents the return statement nodes of the syntax tree

### DECLARATION

---

```
public class ReturnStatement
extends koala.dynamicjava.tree.Statement
implements ExpressionContainer
```

### FIELDS

---

- private Expression expression
  - The expression

### CONSTRUCTORS

---

- *ReturnStatement*

```
public ReturnStatement( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Creates a new while statement
    - **Parameters**
      - \* **exp** - the expression
- 
- *ReturnStatement*

```
public ReturnStatement( koala.dynamicjava.tree.Expression exp,
java.lang.String fn, int bl, int bc, int el, int ec )
```

    - **Usage**
      - \* Creates a new while statement
    - **Parameters**
      - \* **exp** - the expression
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

### METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**

- \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept
- 
- *getExpression*  
 public Expression **getExpression**( )
    - **Usage**
      - \* Gets the expression
- 
- *setExpression*  
 public void **setExpression**( koala.dynamicjava.tree.Expression e )
    - **Usage**
      - \* Sets the expression

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Statement

---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* **visitor** - the visitor to accept
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added

- 
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, boolean **oldValue**, boolean **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

- *getFilename*  
**public String getFilename( )**  
  - **Usage**  
 \* Returns the filename. Can be null.

---
- *getProperties*  
**public Set getProperties( )**  
  - **Usage**  
 \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns the value of a property
  - **Parameters**  
 \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns true if a property is defined for this node
  - **Parameters**  
 \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**  
 \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**  
  - **Usage**  
 \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**



- **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.95 CLASS ShiftLeftAssignExpression

---

This class represents the shift left assign expression nodes of the syntax tree

#### DECLARATION

---

```
public class ShiftLeftAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

#### CONSTRUCTORS

---

- *ShiftLeftAssignExpression*

```
public ShiftLeftAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **lexp** - the LHS expression
      - \* **rexp** - the RHS expression
-

- *ShiftLeftAssignExpression*

```
public ShiftLeftAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression
  - \* **rexp** - the RHS expression
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---

( in 21.2.10, page 1417)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**

- \* Returns the left hand side expression

---

- *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**

- \* Returns the right hand side expression

---

- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

- **Usage**

- \* Sets the left hand side expression

- **Exceptions**
    - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*  
 public void **setRightExpression**( koala.dynamicjava.tree.Expression exp )
    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.
      - \* listener - The PropertyChangeListener to be added
- 
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String propertyName, boolean oldValue, boolean newValue )
    - **Usage**
      - \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.

- **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---
- *getBeginColumn*  
public int **getBeginColumn**( )
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  
public int **getBeginLine**( )
  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  
public int **getEndColumn**( )
  - **Usage**
    - \* Returns the end column of this node in the end line

---
- *getEndLine*  
public int **getEndLine**( )
  - **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  
public String **getFilename**( )
  - **Usage**
    - \* Returns the filename. Can be null.

---
- *getProperties*  
public Set **getProperties**( )

- **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )
  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* name - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* name - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* listener - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* propertyName - The name of the property that was listened on.
    - \* listener - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

## 21.2.96 CLASS ShiftLeftExpression

---

This class represents the shift left expression nodes of the syntax tree

### DECLARATION

---

```
public class ShiftLeftExpression
extends koala.dynamicjava.tree.BinaryExpression
```

### CONSTRUCTORS

---

- *ShiftLeftExpression*  
 public **ShiftLeftExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**  
   \* **lexp** - the LHS expression  
   \* **rexp** - the RHS expression  


---
- *ShiftLeftExpression*  
 public **ShiftLeftExpression**( koala.dynamicjava.tree.Expression lexp,  
 koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int  
 bc, int el, int ec )  
 – **Usage**  
   \* Initializes the expression  
 – **Parameters**

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor **visitor** )

### – Usage

- \* Allows a visitor to traverse the tree

### – Parameters

- \* **visitor** - the visitor to accept

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

- *getLeftExpression*  
 public Expression **getLeftExpression**( )  
 – Usage  
 \* Returns the left hand side expression
- *getRightExpression*  
 public Expression **getRightExpression**( )  
 – Usage  
 \* Returns the right hand side expression
- *setLeftExpression*  
 public void **setLeftExpression**( koala.dynamicjava.tree.Expression **exp** )  
 – Usage  
 \* Sets the left hand side expression  
 – Exceptions  
 \* java.lang.IllegalArgumentException - if exp is null
- *setRightExpression*  
 public void **setRightExpression**( koala.dynamicjava.tree.Expression **exp** )  
 – Usage  
 \* Sets the right hand side expression  
 – Exceptions  
 \* java.lang.IllegalArgumentException - if exp is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.



- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- - *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column
- - *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line
- - *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column
- - *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line
- - *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename
- - *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

## 21.2.97 CLASS ShiftRightAssignExpression

---

This class represents the shift right assign expression nodes of the syntax tree

### DECLARATION

---

```
public class ShiftRightAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

### CONSTRUCTORS

---

- *ShiftRightAssignExpression*  

```
public ShiftRightAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **lexp** - the LHS expression
    - \* **rexp** - the RHS expression

---

- *ShiftRightAssignExpression*  

```
public ShiftRightAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **lexp** - the LHS expression
    - \* **rexp** - the RHS expression
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

### METHODS

---

- *acceptVisitor*  

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.AssignExpression`


---

( in 21.2.10, page 1417)

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.BinaryExpression`


---

( in 21.2.11, page 1421)

- *getLeftExpression*  
`public Expression getLeftExpression( )`  
  - **Usage**  
\* Returns the left hand side expression
- *getRightExpression*  
`public Expression getRightExpression( )`  
  - **Usage**  
\* Returns the right hand side expression
- *setLeftExpression*  
`public void setLeftExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
\* Sets the left hand side expression
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if exp is null
- *setRightExpression*  
`public void setRightExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
\* Sets the right hand side expression
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if exp is null

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`


---

( in 21.2.42, page 1571)

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`


---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
  - **Usage**  
\* Allows a visitor to traverse the tree
  - **Parameters**  
\* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`

- **Usage**
  - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
- **Parameters**
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

- 
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* **propertyName** - The name of the property that was listened on.  
   \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
   \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
   \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
   \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
   \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
   \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

## 21.2.98 CLASS ShiftRightExpression

---

This class represents the shift right expression nodes of the syntax tree

### DECLARATION

---

```
public class ShiftRightExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *ShiftRightExpression*

```
public ShiftRightExpression( koala.dynamicjava.tree.Expression  lexp,
koala.dynamicjava.tree.Expression  rexp )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

• *ShiftRightExpression*

```
public ShiftRightExpression( koala.dynamicjava.tree.Expression  lexp,
koala.dynamicjava.tree.Expression  rexp, java.lang.String  fn, int  bl, int
bc, int  el, int  ec )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

\* **fn** - the filename

\* **bl** - the begin line

\* **bc** - the begin column

\* **el** - the end line

\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

– **Usage**

\* Returns the left hand side expression

• *getRightExpression*

```
public Expression getRightExpression( )
```



- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.

---

\* listener - The PropertyChangeListener to be added

---

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, boolean **oldValue**, boolean **newValue** )

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

public int **getBeginColumn**( )

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

public int **getBeginLine**( )

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

public int **getEndColumn**( )

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

public int **getEndLine**( )

- **Usage**

- \* Returns the end line of this node in the source code

- 
- *getFilename*  
`public String getFilename( )`  
    - **Usage**  
 \* Returns the filename. Can be null.
- 
- *getProperties*  
`public Set getProperties( )`  
    - **Usage**  
 \* Returns the defined properties for this node.
    - **Returns** - a set of string
- 
- *getProperty*  
`public Object getProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns the value of a property
    - **Parameters**  
 \* **name** - the property name
    - **Returns** - null if the property was not previously set
- 
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`  
    - **Usage**  
 \* Returns true if a property is defined for this node
    - **Parameters**  
 \* **name** - the name of the property
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**  
 \* **listener** - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`  
    - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed
- 
- *setBeginColumn*  
`public void setBeginColumn( int i )`  
    - **Usage**  
 \* Sets the begin column
- 
- *setBeginLine*  
`public void setBeginLine( int i )`

- **Usage**
    - \* Sets the begin line
- - *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
- - *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- - *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- - *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.99 CLASS ShortType

---

This class represents the short type nodes of the syntax tree

#### DECLARATION

---

```
public class ShortType
extends koala.dynamicjava.tree.PrimitiveType
```

#### CONSTRUCTORS

---

- *ShortType*

```
public ShortType( )
```

    - **Usage**
      - \* Initializes the type
- - *ShortType*

```
public ShortType( java.lang.String fn, int bl, int bc, int el, int ec )
```

    - **Usage**
      - \* Initializes the type

– **Parameters**

- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimitiveType`

---

( in 21.2.89, page 1795)

• *acceptVisitor*

`public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`

– **Usage**

- \* Allows a visitor to traverse the tree

– **Parameters**

- \* **visitor** - the visitor to accept
- 

• *getValue*

`public Class getValue( )`

– **Usage**

- \* Returns the value of this node
- 

• *setValue*

`public void setValue( java.lang.Class c )`

– **Usage**

- \* Sets the value of this node

– **Exceptions**

- \* `java.lang.IllegalArgumentException` - if `c` is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

• *acceptVisitor*

`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`

– **Usage**

- \* Allows a visitor to traverse the tree

– **Parameters**

- \* **visitor** - the visitor to accept
- 

• *addChangeListener*

`public void addChangeListener( java.beans.PropertyChangeListener listener )`

– **Usage**

- \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code  


---
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line  


---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code  


---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.  


---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string  


---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set  


---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed  


---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* **propertyName** - The name of the property that was listened on.  
   \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
   \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
   \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
   \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
   \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
   \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.100 CLASS SimpleAllocation

---

This class represents the allocation nodes of the syntax tree

#### DECLARATION

---

```
public class SimpleAllocation
extends koala.dynamicjava.tree.Allocation
implements ExpressionStatement
```



FIELDS

---

- public static final String ARGUMENTS
  - The arguments property name
- private List arguments
  - The arguments to pass to the constructor

CONSTRUCTORS

---

- *SimpleAllocation*  
 public **SimpleAllocation**( koala.dynamicjava.tree.Type **tp**, java.util.List **args** )  
    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **tp** - the type prefix
      - \* **args** - the arguments of the constructor
- 
- *SimpleAllocation*  
 public **SimpleAllocation**( koala.dynamicjava.tree.Type **tp**, java.util.List **args**, java.lang.String **fn**, int **bl**, int **bc**, int **el**, int **ec** )  
    - **Usage**
      - \* Initializes the expression
    - **Parameters**
      - \* **tp** - the type prefix
      - \* **args** - the arguments of the constructor
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor **visitor** )  
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* **visitor** - the visitor to accept
- 
- *getArguments*  
 public List **getArguments**( )

---

– **Usage**

\* Returns the constructor arguments

---

• *setArguments*

**public void setArguments( java.util.List l )**

– **Usage**

\* Sets the constructor arguments.

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Allocation`

---

( in 21.2.3, page 1387)

• *getCreationType*

**public Type getCreationType( )**

– **Usage**

\* Returns the creation type

---

• *setCreationType*

**public void setCreationType( koala.dynamicjava.tree.Type t )**

– **Usage**

\* Sets the creation type

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if t is null

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

• *acceptVisitor*

**public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* `visitor` - the visitor to accept

---

• *addPropertyChangeListener*

**public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )**

– **Usage**

- \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added

---
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *getBeginColumn*  

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**  
  - **Usage**  
\* Returns the begin line of this node in the source code

---
- *getEndColumn*  
**public int getEndColumn( )**  
  - **Usage**  
\* Returns the end column of this node in the end line

---
- *getEndLine*  
**public int getEndLine( )**  
  - **Usage**  
\* Returns the end line of this node in the source code

---
- *getFilename*  
**public String getFilename( )**  
  - **Usage**  
\* Returns the filename. Can be null.

---
- *getProperties*  
**public Set getProperties( )**  
  - **Usage**  
\* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**  
  - **Usage**  
\* Returns the value of a property
  - **Parameters**  
\* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**  
  - **Usage**  
\* Returns true if a property is defined for this node
  - **Parameters**  
\* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**  
  - **Usage**  
\* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**  
\* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* **propertyName** - The name of the property that was listened on.  
   \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
   \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
   \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
   \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
   \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
   \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.101 CLASS SimpleAssignExpression

---

This class represents the assign expression nodes of the syntax tree

#### DECLARATION

---

```
public class SimpleAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

CONSTRUCTORS

---

• *SimpleAssignExpression*

```
public SimpleAssignExpression( koala.dynamicjava.tree.Expression lexp,
                               koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

• *SimpleAssignExpression*

```
public SimpleAssignExpression( koala.dynamicjava.tree.Expression lexp,
                               koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
                               bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

\* **fn** - the filename

\* **bl** - the begin line

\* **bc** - the begin column

\* **el** - the end line

\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---

( in 21.2.10, page 1417)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**
  - \* Returns the left hand side expression
- *getRightExpression*


---

```
public Expression getRightExpression( )
```

  - **Usage**
    - \* Returns the right hand side expression
- *setLeftExpression*


---

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

  - **Usage**
    - \* Sets the left hand side expression
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if exp is null
- *setRightExpression*


---

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

  - **Usage**
    - \* Sets the right hand side expression
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if exp is null

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*


---

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added
- *addPropertyChangeListener*


---

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
- **Parameters**
  - \* **propertyName** - The name of the property to listen on.
  - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, boolean **oldValue**, boolean **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
public int **getBeginColumn**( )
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
public int **getBeginLine**( )
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
public int **getEndColumn**( )
  - **Usage**
    - \* Returns the end column of this node in the end line



- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )

- **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.102 CLASS Statement

---

This class represents the statement nodes of the syntax tree

#### DECLARATION

---

```
public abstract class Statement
extends koala.dynamicjava.tree.Node
```

#### CONSTRUCTORS

---

- *Statement*

```
protected Statement( java.lang.String fn, int bl, int bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the statement
  - **Parameters**

- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 — **Usage**  
 \* Allows a visitor to traverse the tree  
 — **Parameters**  
 \* visitor - the visitor to accept

---

- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 — **Usage**  
 \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.  
 — **Parameters**  
 \* listener - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 — **Usage**  
 \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.  
 — **Parameters**  
 \* **propertyName** - The name of the property to listen on.  
 \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String propertyName, boolean oldValue, boolean newValue )  
 — **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 — **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String propertyName, int oldValue, int newValue )  
 — **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String **name** )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**, java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int **i** )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int **i** )
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
 public void **setEndLine**( int **i** )
  - **Usage**
    - \* Sets the end line

---
- *setFilename*  
 public void **setFilename**( java.lang.String **s** )
  - **Usage**

- \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.103 CLASS StaticFieldAccess

---

This class represents the field access nodes of the syntax tree

#### DECLARATION

---

```
public class StaticFieldAccess
extends koala.dynamicjava.tree.FieldAccess
```

#### FIELDS

---

- public static final String FIELD\_TYPE
  - The fieldType property name
- private ReferenceType fieldType
  - The type on which this field access applies

#### CONSTRUCTORS

---

- *StaticFieldAccess*

```
public StaticFieldAccess( koala.dynamicjava.tree.ReferenceType typ,
java.lang.String fln )
```

    - **Usage**
      - \* Creates a new field access node
    - **Parameters**
      - \* **typ** - the type on which this field access applies
      - \* **fln** - the field name
- 
- *StaticFieldAccess*

```
public StaticFieldAccess( koala.dynamicjava.tree.ReferenceType typ,
java.lang.String fln, java.lang.String fn, int bl, int bc, int el, int
ec )
```

    - **Usage**
      - \* Creates a new field access node

– **Parameters**

- \* **typ** - the type on which this field access applies
- \* **fln** - the field name
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

- \* Allows a visitor to traverse the tree

– **Parameters**

- \* **visitor** - the visitor to accept
- 

- *getFieldType*

```
public ReferenceType getFieldType( )
```

– **Usage**

- \* Returns the declaring type of the field
- 

- *setFieldType*

```
public void setFieldType( koala.dynamicjava.tree.ReferenceType t )
```

– **Usage**

- \* Sets the declaring type of the field

– **Exceptions**

- \* `java.lang.IllegalArgumentException` - if t is null

## METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.FieldAccess`

---

( in 21.2.43, page 1575)

- *getFieldName*

```
public String getFieldName( )
```

– **Usage**

- \* Returns the field name
- 

- *setFieldName*

```
public void setFieldName( java.lang.String s )
```

– **Usage**

- \* Sets the field name

## METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**



- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*  
**public int getEndLine( )**
  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*  
**public String getFilename( )**
  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*  
**public Set getProperties( )**
  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---

- *getProperty*  
**public Object getProperty( java.lang.String name )**
  - **Usage**
    - \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*  
`public void setProperty( java.lang.String name, java.lang.Object value )`
    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.104 CLASS StaticMethodCall

---

This class represents the static method call nodes of the syntax tree

#### DECLARATION

---

```
public class StaticMethodCall
extends koala.dynamicjava.tree.MethodCall
```

#### FIELDS

---

- public static final String METHOD\_TYPE
  - The methodType property name
- private ReferenceType methodType
  - The type on which this method call applies

#### CONSTRUCTORS

---

- *StaticMethodCall*  
`public StaticMethodCall( koala.dynamicjava.tree.ReferenceType typ, java.lang.String mn, java.util.List args )`
    - **Usage**
      - \* Creates a new node
    - **Parameters**
      - \* **typ** - the type on which this method call applies
      - \* **mn** - the field name
      - \* **args** - the arguments. Can be null.
- 
- *StaticMethodCall*  
`public StaticMethodCall( koala.dynamicjava.tree.ReferenceType typ, java.lang.String mn, java.util.List args, java.lang.String fn, int bl, int bc, int el, int ec )`
    - **Usage**

- \* Creates a new node
- **Parameters**
  - \* **typ** - the type on which this method call applies
  - \* **mn** - the field name
  - \* **args** - the arguments. Can be null.
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*  
 public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
  
 – **Usage**
  - \* Allows a visitor to traverse the tree
 – **Parameters**
  - \* **visitor** - the visitor to accept

---

- *getMethodType*  
 public ReferenceType **getMethodType**( )  
  
 – **Usage**
  - \* Returns the type on which this method call applies

---

- *setMethodType*  
 public void **setMethodType**( koala.dynamicjava.tree.ReferenceType t )  
  
 – **Usage**
  - \* Sets the declaring type of the method
 – **Exceptions**
  - \* `java.lang.IllegalArgumentException` - if t is null

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.MethodCall

---

( in 21.2.70, page 1706)

- *getArguments*  
 public List **getArguments**( )  
  
 – **Usage**
    - \* Returns the arguments.
 – **Returns** - null if there is no argument

---

  - *getMethodName*  
 public String **getMethodName**( )  
  
 – **Usage**
    - \* Returns the method name
-

- *setArguments*  
 public void **setArguments**( java.util.List l )  
 – **Usage**  
   \* Sets the constructor arguments.

---

- *setMethodName*  
 public void **setMethodName**( java.lang.String s )  
 – **Usage**  
   \* Sets the method name  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if s is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* visitor - the visitor to accept

---

- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener to the listener list.  
   The listener is registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.  
 – **Parameters**

- \* **propertyName** - The name of the property to listen on.
  - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- 
- \* Returns the end line of this node in the source code
- 
- *getFilename*  
**public String getFilename( )**
    - **Usage**
      - \* Returns the filename. Can be null.
- 
- *getProperties*  
**public Set getProperties( )**
    - **Usage**
      - \* Returns the defined properties for this node.
    - **Returns** - a set of string
- 
- *getProperty*  
**public Object getProperty( java.lang.String name )**
    - **Usage**
      - \* Returns the value of a property
    - **Parameters**
      - \* **name** - the property name
    - **Returns** - null if the property was not previously set
- 
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**
    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* **name** - the name of the property
- 
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**
    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be removed
- 
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
    - **Usage**
      - \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**
      - \* **propertyName** - The name of the property that was listened on.
      - \* **listener** - The PropertyChangeListener to be removed
- 
- *setBeginColumn*  
**public void setBeginColumn( int i )**
    - **Usage**
      - \* Sets the begin column
-

- *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
   \* Sets the begin line  


---
- *setEndColumn*  
 public void **setEndColumn**( int i )  
 – **Usage**  
   \* Sets the end column  


---
- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set  


---

### 21.2.105 CLASS StringLiteral

---

This class represents the string literal nodes of the syntax tree

#### DECLARATION

---

```
public class StringLiteral
extends koala.dynamicjava.tree.Literal
```

#### CONSTRUCTORS

---

- *StringLiteral*  
 public **StringLiteral**( java.lang.String rep )  
 – **Usage**  
   \* Initializes a literal  
 – **Parameters**  
   \* **rep** - the representation of the literal  
   \* **val** - the value of this string  


---



- *StringLiteral*

```
public StringLiteral( java.lang.String rep, java.lang.String fn, int bl,
int bc, int el, int ec )
```

- **Usage**

- \* Initializes a literal

- **Parameters**

- \* **rep** - the representation of the literal
    - \* **val** - the value of this string
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

---

- *decodeString*

```
public static String decodeString( java.lang.String rep )
```

- **Usage**

- \* Decodes the representation of a Java literal string.  
The input is not checked since this method always called  
on a string produced by the parser.

- **Parameters**

- \* **rep** - the representation of the character

- **Returns** - the character represented by the given string

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Literal

---

( in 21.2.67, page 1691)

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getRepresentation*

```
public String getRepresentation( )
```

- **Usage**

- \* Returns the representation of this object

---

- *getType*

```
public Class getType( )
```

- **Usage**

- \* Returns the type of this expression.  
NOTE: the 'null' literal has a null type

---

- *getValue*  
 public Object **getValue**( )  
 – **Usage**  
   \* Returns the value of this expression  


---
- *setRepresentation*  
 public void **setRepresentation**( java.lang.String s )  
 – **Usage**  
   \* Sets the representation of this object  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if s is null  


---
- *setType*  
 public void **setType**( java.lang.Class c )  
 – **Usage**  
   \* Sets the type of this object  


---
- *setValue*  
 public void **setValue**( java.lang.Object o )  
 – **Usage**  
   \* Sets the value of this object  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if o is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* visitor - the visitor to accept  


---
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener to the listener list.  
   The listener is registered for all properties.

- **Parameters**
    - \* *listener* - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* *propertyName* - The name of the property to listen on.
    - \* *listener* - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* *propertyName* - The programmatic name of the property that was changed.
    - \* *oldValue* - The old value of the property.
    - \* *newValue* - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

  - **Usage**
    - \* Returns the end line of this node in the source code

---

- *getFilename*

```
public String getFilename( )
```

  - **Usage**
    - \* Returns the filename. Can be null.

---

- *getProperties*

```
public Set getProperties( )
```

  - **Usage**
    - \* Returns the defined properties for this node.
  - **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**

- \* Removes a PropertyChangeListener for a specific property.
- **Parameters**
  - \* **propertyName** - The name of the property that was listened on.
  - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.106 CLASS SubtractAssignExpression

---

This class represents the subtract assign expression nodes of the syntax tree

#### DECLARATION

---

```
public class SubtractAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

CONSTRUCTORS

---

• *SubtractAssignExpression*

```
public SubtractAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression

• *SubtractAssignExpression*

```
public SubtractAssignExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression  
 \* **rexp** - the RHS expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---

( in 21.2.10, page 1417)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**
  - \* Returns the left hand side expression
- *getRightExpression*


---

```
public Expression getRightExpression( )
```

  - **Usage**
    - \* Returns the right hand side expression
- *setLeftExpression*


---

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

  - **Usage**
    - \* Sets the left hand side expression
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if exp is null
- *setRightExpression*


---

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

  - **Usage**
    - \* Sets the right hand side expression
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if exp is null

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*


---

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be added
- *addPropertyChangeListener*


---

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**

- \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---



- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )

- **Usage**
  - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.107 CLASS SubtractExpression

---

This class represents the subtract expression nodes of the syntax tree

#### DECLARATION

---

```
public class SubtractExpression
extends koala.dynamicjava.tree.BinaryExpression
```

#### CONSTRUCTORS

---

- *SubtractExpression*

```
public SubtractExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression

---

- *SubtractExpression*

```
public SubtractExpression( koala.dynamicjava.tree.Expression lexp,
koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int bl, int
bc, int el, int ec )
```

- **Usage**

- \* Initializes the expression

- **Parameters**

- \* **lexp** - the LHS expression
- \* **rexp** - the RHS expression
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

---

## METHODS

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

( in 21.2.11, page 1421)

- *getLeftExpression*

```
public Expression getLeftExpression( )
```

- **Usage**

- \* Returns the left hand side expression

---

- *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**

- \* Returns the right hand side expression

---

- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

- **Usage**

- \* Sets the left hand side expression

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if exp is null
-

- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression  exp )
```

- **Usage**

- \* Sets the right hand side expression

- **Exceptions**

- \* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* visitor - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

- **Parameters**

- \* listener - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName, java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* propertyName - The name of the property to listen on.
  - \* listener - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

- **Parameters**

- \* propertyName - The programmatic name of the property that was changed.

- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
- No event is fired if old and new are equal and non-null.

- **Parameters**

- \* `propertyName` - The programmatic name of the property that was changed.
- \* `oldValue` - The old value of the property.
- \* `newValue` - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code

---

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.

---

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns the value of a property
  - **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int i )
```

- **Usage**
    - \* Sets the end line
- 
- *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.108 CLASS SuperFieldAccess

This class represents the super field access nodes of the syntax tree

#### DECLARATION

```
public class SuperFieldAccess
extends koala.dynamicjava.tree.FieldAccess
```

#### CONSTRUCTORS

- *SuperFieldAccess*

```
public SuperFieldAccess( java.lang.String fln )
```

    - **Usage**
      - \* Creates a new field access node
    - **Parameters**
      - \* **fln** - the field name
- 
- *SuperFieldAccess*

```
public SuperFieldAccess( java.lang.String fln, java.lang.String fn, int
bl, int bc, int el, int ec )
```

    - **Usage**
      - \* Creates a new field access node
    - **Parameters**
      - \* **fln** - the field name
      - \* **fn** - the filename
      - \* **bl** - the begin line
      - \* **bc** - the begin column
      - \* **el** - the end line
      - \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.FieldAccess

---

( in 21.2.43, page 1575)

• *getFieldName*

```
public String getFieldName( )
```

– **Usage**

\* Returns the field name

• *setFieldName*

```
public void setFieldName( java.lang.String  s )
```

– **Usage**

\* Sets the field name

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener )
```



- **Usage**
  - \* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.
- **Parameters**
  - \* `listener` - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*  
**public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**
  - **Usage**
    - \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

- 
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**,  
 java.beans.PropertyChangeListener **listener** )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* **propertyName** - The name of the property that was listened on.  
   \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )  
 – **Usage**  
   \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int **i** )  
 – **Usage**  
   \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int **i** )  
 – **Usage**  
   \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int **i** )  
 – **Usage**  
   \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String **s** )  
 – **Usage**  
   \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String **name**, java.lang.Object **value** )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.109 CLASS SuperMethodCall

---

This class represents the super method call nodes of the syntax tree

#### DECLARATION

---

```
public class SuperMethodCall
extends koala.dynamicjava.tree.MethodCall
```

CONSTRUCTORS

---

• *SuperMethodCall*

```
public SuperMethodCall( java.lang.String mn, java.util.List args )
```

– **Usage**

\* Creates a new node

– **Parameters**

\* **mn** - the method name

\* **args** - the arguments. null if no arguments.

• *SuperMethodCall*

```
public SuperMethodCall( java.lang.String mn, java.util.List args,
java.lang.String fn, int bl, int bc, int el, int ec )
```

– **Usage**

\* Creates a new node

– **Parameters**

\* **mn** - the method name

\* **args** - the arguments. null if no arguments.

\* **fn** - the filename

\* **bl** - the begin line

\* **bc** - the begin column

\* **el** - the end line

\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.MethodCall

---

( in 21.2.70, page 1706)

• *getArguments*

```
public List getArguments( )
```

– **Usage**

\* Returns the arguments.

– **Returns** - null if there is no argument

• *getMethodName*

```
public String getMethodName( )
```

– **Usage**

- \* Returns the method name
- 
- *setArguments*  
 public void **setArguments**( java.util.List l )
    - **Usage**
      - \* Sets the constructor arguments.
- 
- *setMethodName*  
 public void **setMethodName**( java.lang.String s )
    - **Usage**
      - \* Sets the method name
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if s is null

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimaryExpression

---

( in 21.2.88, page 1792)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

---

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*  
**public int getEndLine( )**

- **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  
`public String getFilename( )`
    - **Usage**
      - \* Returns the filename. Can be null.

---
- *getProperties*  
`public Set getProperties( )`
    - **Usage**
      - \* Returns the defined properties for this node.
    - **Returns** - a set of string

---
- *getProperty*  
`public Object getProperty( java.lang.String name )`
    - **Usage**
      - \* Returns the value of a property
    - **Parameters**
      - \* **name** - the property name
    - **Returns** - null if the property was not previously set

---
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`
    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`
    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`
    - **Usage**
      - \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**
      - \* **propertyName** - The name of the property that was listened on.
      - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
`public void setBeginColumn( int i )`
    - **Usage**
      - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
   \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int i )  
 – **Usage**  
   \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.110 CLASS SwitchBlock

---

This class represents the switch expression-statement bindings

#### DECLARATION

---

```
public class SwitchBlock
extends koala.dynamicjava.tree.Node
implements ExpressionContainer
```

#### FIELDS

---

- public static final String STATEMENTS
  - The statements property name
- private Expression expression
  - The expression
- private List statements
  - The statements



CONSTRUCTORS

---

• *SwitchBlock*

```
public SwitchBlock( koala.dynamicjava.tree.Expression exp, java.util.List
stmts )
```

## – Usage

```
* Creates a new binding
```

---

• *SwitchBlock*

```
public SwitchBlock( koala.dynamicjava.tree.Expression exp, java.util.List
stmts, java.lang.String fn, int bl, int bc, int el, int ec )
```

## – Usage

```
* Creates a new binding
```

## – Parameters

```
* fn - the filename
* bl - the begin line
* bc - the begin column
* el - the end line
* ec - the end column
```

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

```
* Allows a visitor to traverse the tree
```

## – Parameters

```
* visitor - the visitor to accept
```

---

• *getExpression*

```
public Expression getExpression( )
```

## – Usage

```
* Returns the 'case' expression
```

---

• *getStatements*

```
public List getStatements( )
```

## – Usage

```
* Returns the statements
```

---

• *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression e )
```

## – Usage

```
* Sets the 'case' expression
```

---

- *setStatements*  
**public void setStatements( java.util.List l )**  
 – **Usage**  
 \* Sets the statements

---

#### METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
**public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**  
 – **Usage**  
 \* Allows a visitor to traverse the tree  
 – **Parameters**  
 \* *visitor* - the visitor to accept
- *addPropertyChangeListener*  
**public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )**  
 – **Usage**  
 \* Adds a PropertyChangeListener to the listener list.  
 The listener is registered for all properties.  
 – **Parameters**  
 \* *listener* - The PropertyChangeListener to be added
- *addPropertyChangeListener*  
**public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**  
 – **Usage**  
 \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.  
 – **Parameters**  
 \* *propertyName* - The name of the property to listen on.  
 \* *listener* - The PropertyChangeListener to be added
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* *propertyName* - The programmatic name of the property that was changed.  
 \* *oldValue* - The old value of the property.  
 \* *newValue* - The new value of the property.
- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**  
 – **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String **name** )
  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String **propertyName**, java.beans.PropertyChangeListener **listener** )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
 public void **setBeginColumn**( int **i** )
  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*  
 public void **setBeginLine**( int **i** )
  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*  
 public void **setEndColumn**( int **i** )
  - **Usage**
    - \* Sets the end column

---
- *setEndLine*  
 public void **setEndLine**( int **i** )
  - **Usage**
    - \* Sets the end line

---
- *setFilename*  
 public void **setFilename**( java.lang.String **s** )
  - **Usage**

- \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.111 CLASS SwitchStatement

---

This class represents the switch statement nodes of the syntax tree

#### DECLARATION

---

```
public class SwitchStatement
extends koala.dynamicjava.tree.Statement
```

#### FIELDS

---

- public static final String SELECTOR
  - The selector property name
- public static final String BINDINGS
  - The bindings property name
- private Expression selector
  - The selector
- private List bindings
  - The list of case bindings

#### CONSTRUCTORS

---

- *SwitchStatement*

```
public SwitchStatement( koala.dynamicjava.tree.Expression  sel,
java.util.List  cases, java.lang.String  fn, int  bl, int  bc, int  el, int
ec )
```

  - **Usage**
    - \* Creates a new switch statement
  - **Parameters**
    - \* **sel** - the selector
    - \* **cases** - the case bindings (SwitchBlocks)
    - \* **fn** - the filename

\* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

\* Allows a visitor to traverse the tree

- **Parameters**

\* **visitor** - the visitor to accept

---

- *getBindings*

```
public List getBindings( )
```

- **Usage**

\* Returns the 'case' bindings

---

- *getSelector*

```
public Expression getSelector( )
```

- **Usage**

\* Gets the selector

---

- *setBindings*

```
public void setBindings( java.util.List  l )
```

- **Usage**

\* Sets the bindings

- **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

---

- *setSelector*

```
public void setSelector( koala.dynamicjava.tree.Expression  e )
```

- **Usage**

\* Sets the selector

- **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

## METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
    java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**



- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- - *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column
- - *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line
- - *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column
- - *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line
- - *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename
- - *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.112 CLASS SynchronizedStatement

---

This class represents the synchronized statement nodes of the syntax tree

#### DECLARATION

---

```
public class SynchronizedStatement
extends koala.dynamicjava.tree.Statement
```

#### FIELDS

---

- public static final String LOCK
  - The lock property name
- public static final String BODY
  - The body property name
- private Expression lock
  - The lock object
- private Node body
  - The body of this statement

#### CONSTRUCTORS

---

- *SynchronizedStatement*  

```
public SynchronizedStatement( koala.dynamicjava.tree.Expression lock,
koala.dynamicjava.tree.Node body, java.lang.String fn, int bl, int bc,
int el, int ec )
```

  - **Usage**
    - \* Creates a new while statement
  - **Parameters**
    - \* **lock** - the lock object
    - \* **body** - the body
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getBody*

```
public Node getBody( )
```

– **Usage**

\* Returns the body of this statement

---

• *getLock*

```
public Expression getLock( )
```

– **Usage**

\* Gets the lock object

---

• *setBody*

```
public void setBody( koala.dynamicjava.tree.Node  node )
```

– **Usage**

\* Sets the body of this statement

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if node is null

---

• *setLock*

```
public void setLock( koala.dynamicjava.tree.Expression  e )
```

– **Usage**

\* Sets the condition to evaluate

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`

---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.113 CLASS ThisExpression

---

This class represents the 'this' expression nodes of the syntax tree

#### DECLARATION

---

```
public class ThisExpression
extends koala.dynamicjava.tree.PrimaryExpression
```

#### FIELDS

---

- public static final String CLASS\_NAME
  - The className property name
- private String className
  - The class that qualify that object

#### CONSTRUCTORS

---

- *ThisExpression*

```
public ThisExpression( java.util.List ids, java.lang.String fn, int bl,
int bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **ids** - the identifiers (tokens) that qualify this 'this'.  
Can be null.
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

#### METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

---

- *getClassName*  
`public String getClassName( )`  
 – **Usage**  
 \* Returns the name of the class that qualify that object
- 

- *setClassName*  
`public void setClassName( java.lang.String s )`  
 – **Usage**  
 \* Sets the name of the class that qualify that object  
 – **Exceptions**  
 \* `java.lang.IllegalArgumentException` - if s is null or body is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
 – **Usage**  
 \* Allows a visitor to traverse the tree  
 – **Parameters**  
 \* `visitor` - the visitor to accept
- 
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`  
 – **Usage**  
 \* Adds a `PropertyChangeListener` to the listener list.  
 The listener is registered for all properties.  
 – **Parameters**  
 \* `listener` - The `PropertyChangeListener` to be added
- 
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`  
 – **Usage**  
 \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.



- **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*  
**protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )**
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*  
**public int getBeginColumn( )**
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---

- *getBeginLine*  
**public int getBeginLine( )**
  - **Usage**
    - \* Returns the begin line of this node in the source code

---

- *getEndColumn*  
**public int getEndColumn( )**
  - **Usage**
    - \* Returns the end column of this node in the end line

---

- *getEndLine*  
**public int getEndLine( )**

- **Usage**
    - \* Returns the end line of this node in the source code

---
- *getFilename*  
`public String getFilename( )`
    - **Usage**
      - \* Returns the filename. Can be null.

---
- *getProperties*  
`public Set getProperties( )`
    - **Usage**
      - \* Returns the defined properties for this node.
    - **Returns** - a set of string

---
- *getProperty*  
`public Object getProperty( java.lang.String name )`
    - **Usage**
      - \* Returns the value of a property
    - **Parameters**
      - \* **name** - the property name
    - **Returns** - null if the property was not previously set

---
- *hasProperty*  
`public boolean hasProperty( java.lang.String name )`
    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* **name** - the name of the property

---
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )`
    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**
      - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
`public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )`
    - **Usage**
      - \* Removes a PropertyChangeListener for a specific property.
    - **Parameters**
      - \* **propertyName** - The name of the property that was listened on.
      - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
`public void setBeginColumn( int i )`
    - **Usage**
      - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )  
 – **Usage**  
   \* Sets the begin line  


---
- *setEndColumn*  
 public void **setEndColumn**( int i )  
 – **Usage**  
   \* Sets the end column  


---
- *setEndLine*  
 public void **setEndLine**( int i )  
 – **Usage**  
   \* Sets the end line  


---
- *setFilename*  
 public void **setFilename**( java.lang.String s )  
 – **Usage**  
   \* Sets the filename  


---
- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )  
 – **Usage**  
   \* Sets the value of a property  
 – **Parameters**  
   \* **name** - the property name  
   \* **value** - the new value to set

### 21.2.114 CLASS ThrowStatement

---

This class represents the throw statement nodes of the syntax tree

#### DECLARATION

---

```
public class ThrowStatement
extends koala.dynamicjava.tree.Statement
implements ExpressionContainer
```

#### FIELDS

---

- private Expression expression  
 – The expression

CONSTRUCTORS

---

• *ThrowStatement*

```
public ThrowStatement( koala.dynamicjava.tree.Expression  exp )
```

– **Usage**

\* Creates a new while statement

– **Parameters**

\* **exp** - the expression

---

• *ThrowStatement*

```
public ThrowStatement( koala.dynamicjava.tree.Expression  exp,  
java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

– **Usage**

\* Creates a new while statement

– **Parameters**

\* **exp** - the expression

\* **fn** - the filename

\* **bl** - the begin line

\* **bc** - the begin column

\* **el** - the end line

\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

• *getExpression*

```
public Expression getExpression( )
```

– **Usage**

\* Gets the expression

---

• *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression  e )
```

– **Usage**

\* Sets the expression

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Statement`

---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

  - **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

\* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

- **Usage**

- \* Sets the value of a property

- **Parameters**

- \* **name** - the property name

- \* **value** - the new value to set

## 21.2.115 CLASS TreeUtilities

---

This class contains a collection of utility methods for trees.

### DECLARATION

---

```
public class TreeUtilities
extends java.lang.Object
```

### CONSTRUCTORS

---

- *TreeUtilities*

```
private TreeUtilities( )
```

- **Usage**

- \* This class contains only static methods, so it is not useful to create instances of it or to extend it.

### METHODS

---

- *classToType*

```
public static Type classToType( java.lang.Class c )
```

- **Usage**

- \* Creates Type node from a Class object

- **Parameters**

- \* **c** - the class to use

---

- *classToType*

```
public static Type classToType( java.lang.Class c, java.lang.String fn,
int bl, int bc, int el, int ec )
```

- **Usage**

- \* Creates Type node from a Class object

- **Parameters**

- \* **c** - the class to use

- \* **fn** - the filename



- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

---

- *listToName*

```
public static String listToName( java.util.List l )
```

- **Usage**

- \* Transforms a list of token into a dot-separated name

- **Parameters**

- \* **l** - a list of token. l can be null.

- **Returns** - "" if l is null.

## 21.2.116 CLASS TryStatement

---

This class represents the try statement nodes of the syntax tree

### DECLARATION

---

```
public class TryStatement
extends koala.dynamicjava.tree.Statement
```

### FIELDS

---

- private Node tryBlock
  - The try block
- private List catchStatements
  - The catch statements
- private Node finallyBlock
  - The finally block

### CONSTRUCTORS

---

- *TryStatement*

```
public TryStatement( koala.dynamicjava.tree.Node tryB, java.util.List
catchL, koala.dynamicjava.tree.Node fn, java.lang.String fn, int bl, int
bc, int el, int ec )
```

- **Usage**

- \* Creates a new while statement

- **Parameters**

- \* **tryB** - the try block
- \* **catchL** - the catch list

- \* **fin** - the finally block
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *getCatchStatements*

```
public List getCatchStatements( )
```

- **Usage**

- \* Gets the catch statements

---

- *getFinallyBlock*

```
public Node getFinallyBlock( )
```

- **Usage**

- \* Gets the finally block

---

- *getTryBlock*

```
public Node getTryBlock( )
```

- **Usage**

- \* Gets the try block

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Statement

---

( in 21.2.102, page 1857)

## METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

\* visitor - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener
)
```

- Usage

- \* Adds a PropertyChangeListener to the listener list.
  - The listener is registered for all properties.

- Parameters

- \* listener - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName,
java.beans.PropertyChangeListener listener )
```

- Usage

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- Parameters

- \* propertyName - The name of the property to listen on.
  - \* listener - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean
oldValue, boolean newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
  - \* oldValue - The old value of the property.
  - \* newValue - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue,
int newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
  - \* oldValue - The old value of the property.
  - \* newValue - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName,
java.lang.Object oldValue, java.lang.Object newValue )
```

- Usage

- \* Report a bound property update to any registered listeners.
  - No event is fired if old and new are equal and non-null.

- Parameters

- \* propertyName - The programmatic name of the property that was changed.
  - \* oldValue - The old value of the property.
  - \* newValue - The new value of the property.

- 
- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
   \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
   \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
   \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
   \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
   \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
   \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns the value of a property  
 – **Parameters**  
   \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns true if a property is defined for this node  
 – **Parameters**  
   \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**

- \* Removes a `PropertyChangeListener` from the listener list.  
This removes a `PropertyChangeListener` that was registered for all properties.
  - **Parameters**
    - \* `listener` - The `PropertyChangeListener` to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a `PropertyChangeListener` for a specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property that was listened on.
    - \* `listener` - The `PropertyChangeListener` to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* `name` - the property name
    - \* `value` - the new value to set

### 21.2.117 CLASS Type

---

This class represents the type nodes of the syntax tree

DECLARATION

---

```
public abstract class Type
extends koala.dynamicjava.tree.Node
```

CONSTRUCTORS

---

- *Type*  

```
protected Type( java.lang.String  fn, int  bl, int  bc, int  el, int  ec )
```

  - **Usage**
    - \* Initializes the type
  - **Parameters**
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName, java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property to listen on.
    - \* **listener** - The PropertyChangeListener to be added

- 
- *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, boolean **oldValue**, boolean **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, int **oldValue**, int **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *firePropertyChange*  
 protected void **firePropertyChange**( java.lang.String **propertyName**, java.lang.Object **oldValue**, java.lang.Object **newValue** )  
 – **Usage**  
 \* Report a bound property update to any registered listeners.  
 No event is fired if old and new are equal and non-null.  
 – **Parameters**  
 \* **propertyName** - The programmatic name of the property that was changed.  
 \* **oldValue** - The old value of the property.  
 \* **newValue** - The new value of the property.

---

  - *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line

---

  - *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code

---

  - *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line

---

  - *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code

---

- *getFilename*  
**public String getFilename( )**  
  - **Usage**  
 \* Returns the filename. Can be null.

---
- *getProperties*  
**public Set getProperties( )**  
  - **Usage**  
 \* Returns the defined properties for this node.
  - **Returns** - a set of string

---
- *getProperty*  
**public Object getProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns the value of a property
  - **Parameters**  
 \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*  
**public boolean hasProperty( java.lang.String name )**  
  - **Usage**  
 \* Returns true if a property is defined for this node
  - **Parameters**  
 \* **name** - the name of the property

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**  
 \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
**public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**  
  - **Usage**  
 \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**  
 \* **propertyName** - The name of the property that was listened on.  
 \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*  
**public void setBeginColumn( int i )**  
  - **Usage**  
 \* Sets the begin column

---
- *setBeginLine*  
**public void setBeginLine( int i )**



- **Usage**
    - \* Sets the begin line
- - *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
- - *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- - *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- - *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.118 CLASS *TypeDeclaration*

This class represents a type declaration

#### DECLARATION

```
public abstract class TypeDeclaration
extends koala.dynamicjava.tree.Node
```

#### FIELDS

- public static final String ACCESS\_FLAGS
  - The accessFlags property name
- public static final String NAME
  - The name property name
- public static final String INTERFACES
  - The interfaces property name
- public static final String MEMBERS

- The members property name
- private int accessFlags
  - The access flags
- private String name
  - The name of this class
- private List interfaces
  - The implemented interfaces
- private List members
  - The members

## CONSTRUCTORS

---

- *TypeDeclaration*  
 protected **TypeDeclaration**( int flags, java.lang.String name,  
 java.util.List impl, java.util.List body, java.lang.String fn, int bl,  
 int bc, int el, int ec )
  - **Usage**
    - \* Creates a new class declaration
  - **Parameters**
    - \* **flags** - the access flags
    - \* **name** - the name of the class to declare
    - \* **impl** - the list of implemented interfaces (List of Token). Can be null.
    - \* **body** - the list of fields declarations
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

## METHODS

---

- *getAccessFlags*  
 public int **getAccessFlags**( )
    - **Usage**
      - \* Returns the access flags for this class
  - *getInterfaces*  
 public List **getInterfaces**( )
    - **Usage**
      - \* Returns a list that contains the names (String) of the implemented interfaces.  
 Can be null.
-

- *getMembers*  
`public List getMembers( )`  
  - **Usage**  
\* Returns the list of the declared members

---
- *getName*  
`public String getName( )`  
  - **Usage**  
\* Returns the name of this class

---
- *setAccessFlags*  
`public void setAccessFlags( int f )`  
  - **Usage**  
\* Sets the access flags for this constructor

---
- *setInterfaces*  
`public void setInterfaces( java.util.List l )`  
  - **Usage**  
\* Sets the interfaces (a list of strings)

---
- *setMembers*  
`public void setMembers( java.util.List l )`  
  - **Usage**  
\* Sets the members
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if l is null

---
- *setName*  
`public void setName( java.lang.String s )`  
  - **Usage**  
\* Sets the type's name
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if s is null

---

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
  - **Usage**  
\* Allows a visitor to traverse the tree
  - **Parameters**  
\* `visitor` - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.
    - The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added
- 

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
-

- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line  


---
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code  


---
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line  


---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code  


---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.  


---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string  


---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set  


---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.

- **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.119 CLASS TypeExpression

---

This class represents the type expression nodes of the syntax tree

DECLARATION

---

```
public class TypeExpression
extends koala.dynamicjava.tree.PrimaryExpression
```

FIELDS

---

- public static final String TYPE
  - The type property name
- private Type type
  - The type represented by this expression

CONSTRUCTORS

---

- *TypeExpression*  
public **TypeExpression**( koala.dynamicjava.tree.Type t )
  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* t - the type represented by this expression
- *TypeExpression*  
public **TypeExpression**( koala.dynamicjava.tree.Type t, java.lang.String fn, int bl, int bc, int el, int ec )
  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* t - the type represented by this expression
    - \* fn - the filename
    - \* bl - the begin line
    - \* bc - the begin column
    - \* el - the end line
    - \* ec - the end column

METHODS

---

- *acceptVisitor*  
public Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )
  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**

\* **visitor** - the visitor to accept

---

- *getType*

**public Type getType( )**

- **Usage**

\* Returns the type represented by this expression

---

- *setType*

**public void setType( koala.dynamicjava.tree.ReferenceType t )**

- **Usage**

\* Sets the type

- **Exceptions**

\* `java.lang.IllegalArgumentException` - if t is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.PrimaryExpression`

---

( in 21.2.88, page 1792)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*

**public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**

- **Usage**

\* Allows a visitor to traverse the tree

- **Parameters**

\* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

**public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )**

- **Usage**

\* Adds a `PropertyChangeListener` to the listener list.  
The listener is registered for all properties.

- **Parameters**

\* **listener** - The `PropertyChangeListener` to be added

---

- *addPropertyChangeListener*

**public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )**

- **Usage**



- \* Adds a `PropertyChangeListener` for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The `PropertyChangeListener` to be added

---
- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *firePropertyChange*  
`protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )`
  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.

---
- *getBeginColumn*  
`public int getBeginColumn( )`
  - **Usage**
    - \* Returns the begin column of this node in the begin line

---
- *getBeginLine*  
`public int getBeginLine( )`
  - **Usage**
    - \* Returns the begin line of this node in the source code

---
- *getEndColumn*  
`public int getEndColumn( )`
  - **Usage**
    - \* Returns the end column of this node in the end line

- 
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
   \* Returns the end line of this node in the source code

---

  - *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
   \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
   \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns the value of a property  
 – **Parameters**  
   \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
   \* Returns true if a property is defined for this node  
 – **Parameters**  
   \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener from the listener list.  
   This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
   \* propertyName - The name of the property that was listened on.  
   \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )

- **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**
    - \* Sets the filename

---
- *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.120 CLASS *UnaryExpression*

---

This class represents the unary expression nodes of the syntax tree

#### DECLARATION

---

```
public abstract class UnaryExpression
extends koala.dynamicjava.tree.Expression
implements ExpressionContainer
```

#### FIELDS

---

- private Expression expression
  - The target expression

CONSTRUCTORS

---

• *UnaryExpression*

```
protected UnaryExpression( koala.dynamicjava.tree.Expression exp,
    java.lang.String fn, int bl, int bc, int el, int ec )
```

– **Usage**

\* Initializes the expression

– **Parameters**

\* **exp** - the target expression  
 \* **fn** - the filename  
 \* **bl** - the begin line  
 \* **bc** - the begin column  
 \* **el** - the end line  
 \* **ec** - the end column

METHODS

---

• *getExpression*

```
public Expression getExpression( )
```

– **Usage**

\* Returns the target expression

• *setExpression*

```
public void setExpression( koala.dynamicjava.tree.Expression e )
```

– **Usage**

\* Sets the target expression

– **Exceptions**

\* `java.lang.IllegalArgumentException` - if e is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor
    )
```

– **Usage**

\* Allows a visitor to traverse the tree

– **Parameters**

\* **visitor** - the visitor to accept

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener to the listener list.
    - The listener is registered for all properties.

- **Parameters**

- \* **listener** - The PropertyChangeListener to be added
- 

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, boolean oldValue, boolean newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, int oldValue, int newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String propertyName, java.lang.Object oldValue, java.lang.Object newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
-

- *getBeginColumn*  
 public int **getBeginColumn**( )  
 – **Usage**  
 \* Returns the begin column of this node in the begin line  


---
- *getBeginLine*  
 public int **getBeginLine**( )  
 – **Usage**  
 \* Returns the begin line of this node in the source code  


---
- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line  


---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code  


---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.  


---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string  


---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set  


---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.

- **Parameters**
  - \* **listener** - The PropertyChangeListener to be removed

---

- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column

---

- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line

---

- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename

---

- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.121 CLASS UnsignedShiftRightAssignExpression

---

This class represents the unsigned shift right assign expression nodes of the syntax tree

DECLARATION

---

```
public class UnsignedShiftRightAssignExpression
extends koala.dynamicjava.tree.AssignExpression
```

CONSTRUCTORS

---

- *UnsignedShiftRightAssignExpression*  

```
public UnsignedShiftRightAssignExpression(
koala.dynamicjava.tree.Expression lexp, koala.dynamicjava.tree.Expression
rexp )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **lexp** - the LHS expression
    - \* **rexp** - the RHS expression

---

- *UnsignedShiftRightAssignExpression*  

```
public UnsignedShiftRightAssignExpression(
koala.dynamicjava.tree.Expression lexp, koala.dynamicjava.tree.Expression
rexp, java.lang.String fn, int bl, int bc, int el, int ec )
```

  - **Usage**
    - \* Initializes the expression
  - **Parameters**
    - \* **lexp** - the LHS expression
    - \* **rexp** - the RHS expression
    - \* **fn** - the filename
    - \* **bl** - the begin line
    - \* **bc** - the begin column
    - \* **el** - the end line
    - \* **ec** - the end column

METHODS

---

- *acceptVisitor*  

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.AssignExpression

---



METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.BinaryExpression`

---

( in 21.2.11, page 1421)

- *getLeftExpression*  
`public Expression getLeftExpression( )`  
  - **Usage**  
\* Returns the left hand side expression
- *getRightExpression*  
`public Expression getRightExpression( )`  
  - **Usage**  
\* Returns the right hand side expression
- *setLeftExpression*  
`public void setLeftExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
\* Sets the left hand side expression
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if exp is null
- *setRightExpression*  
`public void setRightExpression( koala.dynamicjava.tree.Expression exp )`  
  - **Usage**  
\* Sets the right hand side expression
  - **Exceptions**  
\* `java.lang.IllegalArgumentException` - if exp is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Expression`

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
`public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )`  
  - **Usage**  
\* Allows a visitor to traverse the tree
  - **Parameters**  
\* `visitor` - the visitor to accept
- *addPropertyChangeListener*  
`public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )`  
  - **Usage**  
\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**

\* **listener** - The PropertyChangeListener to be added

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
- \* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---
- *getEndColumn*  
 public int **getEndColumn**( )
    - **Usage**
      - \* Returns the end column of this node in the end line

---
- *getEndLine*  
 public int **getEndLine**( )
    - **Usage**
      - \* Returns the end line of this node in the source code

---
- *getFilename*  
 public String **getFilename**( )
    - **Usage**
      - \* Returns the filename. Can be null.

---
- *getProperties*  
 public Set **getProperties**( )
    - **Usage**
      - \* Returns the defined properties for this node.
    - **Returns** - a set of string

---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )
    - **Usage**
      - \* Returns the value of a property
    - **Parameters**
      - \* name - the property name
    - **Returns** - null if the property was not previously set

---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )
    - **Usage**
      - \* Returns true if a property is defined for this node
    - **Parameters**
      - \* name - the name of the property

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
    - **Usage**
      - \* Removes a PropertyChangeListener for a specific property.

- **Parameters**
  - \* **propertyName** - The name of the property that was listened on.
  - \* **listener** - The PropertyChangeListener to be removed

---

- *setBeginColumn*  
 public void **setBeginColumn**( int i )
  - **Usage**
    - \* Sets the begin column

---

- *setBeginLine*  
 public void **setBeginLine**( int i )
  - **Usage**
    - \* Sets the begin line

---

- *setEndColumn*  
 public void **setEndColumn**( int i )
  - **Usage**
    - \* Sets the end column

---

- *setEndLine*  
 public void **setEndLine**( int i )
  - **Usage**
    - \* Sets the end line

---

- *setFilename*  
 public void **setFilename**( java.lang.String s )
  - **Usage**
    - \* Sets the filename

---

- *setProperty*  
 public void **setProperty**( java.lang.String name, java.lang.Object value )
  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.122 CLASS UnsignedShiftRightExpression

---

This class represents the unsigned shift right expression nodes of the syntax tree

#### DECLARATION

---

```
public class UnsignedShiftRightExpression
extends koala.dynamicjava.tree.BinaryExpression
```

CONSTRUCTORS

---

• *UnsignedShiftRightExpression*

```
public UnsignedShiftRightExpression( koala.dynamicjava.tree.Expression
lexp, koala.dynamicjava.tree.Expression rexp )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

• *UnsignedShiftRightExpression*

```
public UnsignedShiftRightExpression( koala.dynamicjava.tree.Expression
lexp, koala.dynamicjava.tree.Expression rexp, java.lang.String fn, int
bl, int bc, int el, int ec )
```

## – Usage

\* Initializes the expression

## – Parameters

\* **lexp** - the LHS expression

\* **rexp** - the RHS expression

\* **fn** - the filename

\* **bl** - the begin line

\* **bc** - the begin column

\* **el** - the end line

\* **ec** - the end column

METHODS

---

• *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* **visitor** - the visitor to accept

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.BinaryExpression

---

( in 21.2.11, page 1421)

• *getLeftExpression*

```
public Expression getLeftExpression( )
```

## – Usage

\* Returns the left hand side expression

• *getRightExpression*

```
public Expression getRightExpression( )
```

- **Usage**
    - \* Returns the right hand side expression
- 
- *setLeftExpression*

```
public void setLeftExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the left hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null
- 
- *setRightExpression*

```
public void setRightExpression( koala.dynamicjava.tree.Expression exp )
```

    - **Usage**
      - \* Sets the right hand side expression
    - **Exceptions**
      - \* java.lang.IllegalArgumentException - if exp is null

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Expression

---

( in 21.2.42, page 1571)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

    - **Usage**
      - \* Allows a visitor to traverse the tree
    - **Parameters**
      - \* visitor - the visitor to accept
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
    - **Parameters**
      - \* listener - The PropertyChangeListener to be added
- 
- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

    - **Usage**
      - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.
    - **Parameters**
      - \* propertyName - The name of the property to listen on.

---

\* **listener** - The PropertyChangeListener to be added

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
    - \* **oldValue** - The old value of the property.
    - \* **newValue** - The new value of the property.

---

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line

---

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code

---

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line

---

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code

- 
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.

---

  - *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string

---

  - *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set

---

  - *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed

---

  - *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**  
 \* propertyName - The name of the property that was listened on.  
 \* listener - The PropertyChangeListener to be removed

---

  - *setBeginColumn*  
 public void **setBeginColumn**( int i )  
 – **Usage**  
 \* Sets the begin column

---

  - *setBeginLine*  
 public void **setBeginLine**( int i )



- **Usage**
    - \* Sets the begin line
- - *setEndColumn*

```
public void setEndColumn( int i )
```

    - **Usage**
      - \* Sets the end column
- - *setEndLine*

```
public void setEndLine( int i )
```

    - **Usage**
      - \* Sets the end line
- - *setFilename*

```
public void setFilename( java.lang.String s )
```

    - **Usage**
      - \* Sets the filename
- - *setProperty*

```
public void setProperty( java.lang.String name, java.lang.Object value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.123 CLASS VariableDeclaration

---

This class represents variable declarations in an AST

#### DECLARATION

---

```
public class VariableDeclaration
extends koala.dynamicjava.tree.Node
```

#### FIELDS

---

- public static final String FINAL
  - The final property name
- public static final String TYPE
  - The type property name
- public static final String NAME
  - The name property name
- public static final String INITIALIZER
  - The initializer property name

- private boolean finalVariable
  - Whether this variable is final
- private Type type
  - The type of this variable
- private String name
  - The name of this variable
- private Expression initializer
  - The initializer

## CONSTRUCTORS

---

- *VariableDeclaration*

```
public VariableDeclaration( boolean fin, koala.dynamicjava.tree.Type type,
    java.lang.String name, koala.dynamicjava.tree.Expression init )
```

- **Usage**

- \* Creates a new variable declaration

- **Parameters**

- \* **fin** - is this variable final?
  - \* **type** - the type of this variable
  - \* **name** - the name of this variable
  - \* **init** - the initializer

---

- *VariableDeclaration*

```
public VariableDeclaration( boolean fin, koala.dynamicjava.tree.Type type,
    java.lang.String name, koala.dynamicjava.tree.Expression init,
    java.lang.String fn, int bl, int bc, int el, int ec )
```

- **Usage**

- \* Creates a new variable declaration

- **Parameters**

- \* **fin** - is this variable final?
  - \* **type** - the type of this variable
  - \* **name** - the name of this variable
  - \* **init** - the initializer
  - \* **fn** - the filename
  - \* **bl** - the begin line
  - \* **bc** - the begin column
  - \* **el** - the end line
  - \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* **visitor** - the visitor to accept

---
- *getInitializer*

```
public Expression getInitializer( )
```

  - **Usage**
    - \* Returns the initializer for this variable

---
- *getName*

```
public String getName( )
```

  - **Usage**
    - \* Returns the name of this variable

---
- *getType*

```
public Type getType( )
```

  - **Usage**
    - \* Gets the declared type for this variable

---
- *isFinal*

```
public boolean isFinal( )
```

  - **Usage**
    - \* Returns true if this variable is final

---
- *setFinal*

```
public void setFinal( boolean b )
```

  - **Usage**
    - \* Sets the final flag

---
- *setInitializer*

```
public void setInitializer( koala.dynamicjava.tree.Expression e )
```

  - **Usage**
    - \* Sets the initializer

---
- *setName*

```
public void setName( java.lang.String s )
```

  - **Usage**
    - \* Sets the variable's name
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if s is null

---
- *setType*

```
public void setType( koala.dynamicjava.tree.Type t )
```

  - **Usage**
    - \* Sets the type of this field
  - **Exceptions**
    - \* `java.lang.IllegalArgumentException` - if t is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

( in 21.2.75, page 1732)

• *acceptVisitor*

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

## – Usage

\* Allows a visitor to traverse the tree

## – Parameters

\* `visitor` - the visitor to accept

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

## – Usage

\* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.

## – Parameters

\* `listener` - The PropertyChangeListener to be added

• *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

## – Usage

\* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.

## – Parameters

\* `propertyName` - The name of the property to listen on.  
\* `listener` - The PropertyChangeListener to be added

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

## – Usage

\* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

## – Parameters

\* `propertyName` - The programmatic name of the property that was changed.  
\* `oldValue` - The old value of the property.  
\* `newValue` - The new value of the property.

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners.
    - No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
- 

- *getEndColumn*

```
public int getEndColumn( )
```

- **Usage**

- \* Returns the end column of this node in the end line
- 

- *getEndLine*

```
public int getEndLine( )
```

- **Usage**

- \* Returns the end line of this node in the source code
- 

- *getFilename*

```
public String getFilename( )
```

- **Usage**

- \* Returns the filename. Can be null.
- 

- *getProperties*

```
public Set getProperties( )
```

- **Usage**

- \* Returns the defined properties for this node.

- **Returns** - a set of string

---

- *getProperty*

```
public Object getProperty( java.lang.String  name )
```

- **Usage**

- \* Returns the value of a property

- **Parameters**

- \* **name** - the property name

- **Returns** - null if the property was not previously set

---

- *hasProperty*

```
public boolean hasProperty( java.lang.String  name )
```

- **Usage**

- \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener
listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered  
for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed
- *setBeginColumn*

```
public void setBeginColumn( int  i )
```

  - **Usage**
    - \* Sets the begin column
- *setBeginLine*

```
public void setBeginLine( int  i )
```

  - **Usage**
    - \* Sets the begin line
- *setEndColumn*

```
public void setEndColumn( int  i )
```

  - **Usage**
    - \* Sets the end column
- *setEndLine*

```
public void setEndLine( int  i )
```

  - **Usage**
    - \* Sets the end line
- *setFilename*

```
public void setFilename( java.lang.String  s )
```

  - **Usage**
    - \* Sets the filename
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

  - **Usage**
    - \* Sets the value of a property
  - **Parameters**
    - \* **name** - the property name
    - \* **value** - the new value to set

### 21.2.124 CLASS VoidType

---

This class represents the void type nodes of the syntax tree

#### DECLARATION

---

```
public class VoidType
extends koala.dynamicjava.tree.PrimitiveType
```

#### CONSTRUCTORS

---

- *VoidType*  
**public VoidType( )**  
 – **Usage**  
   \* Initializes the type

---

- *VoidType*  
**public VoidType( java.lang.String fn, int bl, int bc, int el, int ec )**  
 – **Usage**  
   \* Initializes the type  
 – **Parameters**  
   \* **fn** - the filename  
   \* **bl** - the begin line  
   \* **bc** - the begin column  
   \* **el** - the end line  
   \* **ec** - the end column

#### METHODS INHERITED FROM CLASS koala.dynamicjava.tree.PrimitiveType

---

( in 21.2.89, page 1795)

- *acceptVisitor*  
**public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )**  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* **visitor** - the visitor to accept

---

- *getValue*  
**public Class getValue( )**  
 – **Usage**  
   \* Returns the value of this node

---

- *setValue*  
**public void setValue( java.lang.Class c )**  
 – **Usage**  
   \* Sets the value of this node  
 – **Exceptions**  
   \* **java.lang.IllegalArgumentException** - if c is null

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Type`

---

( in 21.2.117, page 1924)

METHODS INHERITED FROM CLASS `koala.dynamicjava.tree.Node`

---

( in 21.2.75, page 1732)

- *acceptVisitor*  

```
public abstract Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor  visitor
)
```

  - **Usage**
    - \* Allows a visitor to traverse the tree
  - **Parameters**
    - \* `visitor` - the visitor to accept
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.beans.PropertyChangeListener  listener
)
```

  - **Usage**
    - \* Adds a PropertyChangeListener to the listener list.  
The listener is registered for all properties.
  - **Parameters**
    - \* `listener` - The PropertyChangeListener to be added
- *addPropertyChangeListener*  

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

  - **Usage**
    - \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on `firePropertyChange` names that specific property.
  - **Parameters**
    - \* `propertyName` - The name of the property to listen on.
    - \* `listener` - The PropertyChangeListener to be added
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

  - **Usage**
    - \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.
  - **Parameters**
    - \* `propertyName` - The programmatic name of the property that was changed.
    - \* `oldValue` - The old value of the property.
    - \* `newValue` - The new value of the property.
- *firePropertyChange*  

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

  - **Usage**



- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

– **Usage**

- \* Report a bound property update to any registered listeners.  
No event is fired if old and new are equal and non-null.

– **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
- \* **oldValue** - The old value of the property.
- \* **newValue** - The new value of the property.

---

• *getBeginColumn*

```
public int getBeginColumn( )
```

– **Usage**

- \* Returns the begin column of this node in the begin line

---

• *getBeginLine*

```
public int getBeginLine( )
```

– **Usage**

- \* Returns the begin line of this node in the source code

---

• *getEndColumn*

```
public int getEndColumn( )
```

– **Usage**

- \* Returns the end column of this node in the end line

---

• *getEndLine*

```
public int getEndLine( )
```

– **Usage**

- \* Returns the end line of this node in the source code

---

• *getFilename*

```
public String getFilename( )
```

– **Usage**

- \* Returns the filename. Can be null.

---

• *getProperties*

```
public Set getProperties( )
```

– **Usage**

- \* Returns the defined properties for this node.

– **Returns** - a set of string

---

• *getProperty*

```
public Object getProperty( java.lang.String  name )
```

– **Usage**

- \* Returns the value of a property

- **Parameters**
    - \* **name** - the property name
  - **Returns** - null if the property was not previously set

---
- *hasProperty*

```
public boolean hasProperty( java.lang.String name )
```

  - **Usage**
    - \* Returns true if a property is defined for this node
  - **Parameters**
    - \* **name** - the name of the property

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener from the listener list.  
This removes a PropertyChangeListener that was registered for all properties.
  - **Parameters**
    - \* **listener** - The PropertyChangeListener to be removed

---
- *removePropertyChangeListener*

```
public void removePropertyChangeListener( java.lang.String propertyName, java.beans.PropertyChangeListener listener )
```

  - **Usage**
    - \* Removes a PropertyChangeListener for a specific property.
  - **Parameters**
    - \* **propertyName** - The name of the property that was listened on.
    - \* **listener** - The PropertyChangeListener to be removed

---
- *setBeginColumn*

```
public void setBeginColumn( int i )
```

  - **Usage**
    - \* Sets the begin column

---
- *setBeginLine*

```
public void setBeginLine( int i )
```

  - **Usage**
    - \* Sets the begin line

---
- *setEndColumn*

```
public void setEndColumn( int i )
```

  - **Usage**
    - \* Sets the end column

---
- *setEndLine*

```
public void setEndLine( int i )
```

  - **Usage**
    - \* Sets the end line

---
- *setFilename*

```
public void setFilename( java.lang.String s )
```

  - **Usage**

- \* Sets the filename
- 
- *setProperty*

```
public void setProperty( java.lang.String  name, java.lang.Object  value )
```

    - **Usage**
      - \* Sets the value of a property
    - **Parameters**
      - \* **name** - the property name
      - \* **value** - the new value to set

### 21.2.125 CLASS WhileStatement

---

This class represents the while statement nodes of the syntax tree

#### DECLARATION

---

```
public class WhileStatement
extends koala.dynamicjava.tree.Statement
implements ContinueTarget
```

#### FIELDS

---

- public static final String CONDITION
  - The condition property name
- public static final String BODY
  - The body property name
- private Expression condition
  - The condition to evaluate at each loop
- private Node body
  - The body of this statement
- private List labels
  - The labels

#### CONSTRUCTORS

---

- *WhileStatement*

```
public WhileStatement( koala.dynamicjava.tree.Expression  cond,
koala.dynamicjava.tree.Node  body )
```

  - **Usage**
    - \* Creates a new while statement
  - **Parameters**

- \* **cond** - the condition to evaluate at each loop
- \* **body** - the body

---

- *WhileStatement*

```
public WhileStatement( koala.dynamicjava.tree.Expression cond,
koala.dynamicjava.tree.Node body, java.lang.String fn, int bl, int bc,
int el, int ec )
```

- **Usage**

- \* Creates a new while statement

- **Parameters**

- \* **cond** - the condition to evaluate at each loop
- \* **body** - the body
- \* **fn** - the filename
- \* **bl** - the begin line
- \* **bc** - the begin column
- \* **el** - the end line
- \* **ec** - the end column

## METHODS

---

- *acceptVisitor*

```
public Object acceptVisitor( koala.dynamicjava.tree.visitor.Visitor visitor )
```

- **Usage**

- \* Allows a visitor to traverse the tree

- **Parameters**

- \* **visitor** - the visitor to accept

---

- *addLabel*

```
public void addLabel( java.lang.String label )
```

- **Usage**

- \* Adds a label to this statement

- **Parameters**

- \* **label** - the label to add

- **Exceptions**

- \* `java.lang.IllegalArgumentException` - if label is null

---

- *getBody*

```
public Node getBody( )
```

- **Usage**

- \* Returns the body of this statement

---

- *getCondition*

```
public Expression getCondition( )
```

- **Usage**

- \* Gets the condition to evaluate at each loop

- 
- *hasLabel*  
 public boolean **hasLabel**( java.lang.String label )  
 – **Usage**  
   \* Test whether this statement has the given label  
 – **Returns** - true if this statement has the given label
  - *setBody*  
 public void **setBody**( koala.dynamicjava.tree.Node node )  
 – **Usage**  
   \* Sets the body of this statement  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if node is null
  - *setCondition*  
 public void **setCondition**( koala.dynamicjava.tree.Expression e )  
 – **Usage**  
   \* Sets the condition to evaluate  
 – **Exceptions**  
   \* java.lang.IllegalArgumentException - if e is null
- 

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Statement

---

( in 21.2.102, page 1857)

METHODS INHERITED FROM CLASS koala.dynamicjava.tree.Node

---

( in 21.2.75, page 1732)

- *acceptVisitor*  
 public abstract Object **acceptVisitor**( koala.dynamicjava.tree.visitor.Visitor visitor )  
 – **Usage**  
   \* Allows a visitor to traverse the tree  
 – **Parameters**  
   \* visitor - the visitor to accept
  - *addPropertyChangeListener*  
 public void **addPropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
   \* Adds a PropertyChangeListener to the listener list.  
   The listener is registered for all properties.  
 – **Parameters**  
   \* listener - The PropertyChangeListener to be added
-

---

- *addPropertyChangeListener*

```
public void addPropertyChangeListener( java.lang.String  propertyName,
java.beans.PropertyChangeListener  listener )
```

- **Usage**

- \* Adds a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property.

- **Parameters**

- \* **propertyName** - The name of the property to listen on.
      - \* **listener** - The PropertyChangeListener to be added
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, boolean
oldValue, boolean  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName, int  oldValue,
int  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *firePropertyChange*

```
protected void firePropertyChange( java.lang.String  propertyName,
java.lang.Object  oldValue, java.lang.Object  newValue )
```

- **Usage**

- \* Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

- **Parameters**

- \* **propertyName** - The programmatic name of the property that was changed.
      - \* **oldValue** - The old value of the property.
      - \* **newValue** - The new value of the property.
- 

- *getBeginColumn*

```
public int getBeginColumn( )
```

- **Usage**

- \* Returns the begin column of this node in the begin line
- 

- *getBeginLine*

```
public int getBeginLine( )
```

- **Usage**

- \* Returns the begin line of this node in the source code
-

- *getEndColumn*  
 public int **getEndColumn**( )  
 – **Usage**  
 \* Returns the end column of this node in the end line  


---
- *getEndLine*  
 public int **getEndLine**( )  
 – **Usage**  
 \* Returns the end line of this node in the source code  


---
- *getFilename*  
 public String **getFilename**( )  
 – **Usage**  
 \* Returns the filename. Can be null.  


---
- *getProperties*  
 public Set **getProperties**( )  
 – **Usage**  
 \* Returns the defined properties for this node.  
 – **Returns** - a set of string  


---
- *getProperty*  
 public Object **getProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns the value of a property  
 – **Parameters**  
 \* name - the property name  
 – **Returns** - null if the property was not previously set  


---
- *hasProperty*  
 public boolean **hasProperty**( java.lang.String name )  
 – **Usage**  
 \* Returns true if a property is defined for this node  
 – **Parameters**  
 \* name - the name of the property  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener from the listener list.  
 This removes a PropertyChangeListener that was registered for all properties.  
 – **Parameters**  
 \* listener - The PropertyChangeListener to be removed  


---
- *removePropertyChangeListener*  
 public void **removePropertyChangeListener**( java.lang.String propertyName, java.beans.PropertyChangeListener listener )  
 – **Usage**  
 \* Removes a PropertyChangeListener for a specific property.  
 – **Parameters**

- \* **propertyName** - The name of the property that was listened on.
  - \* **listener** - The PropertyChangeListener to be removed
- 

- *setBeginColumn*

public void **setBeginColumn**( int i )

- **Usage**

- \* Sets the begin column
- 

- *setBeginLine*

public void **setBeginLine**( int i )

- **Usage**

- \* Sets the begin line
- 

- *setEndColumn*

public void **setEndColumn**( int i )

- **Usage**

- \* Sets the end column
- 

- *setEndLine*

public void **setEndLine**( int i )

- **Usage**

- \* Sets the end line
- 

- *setFilename*

public void **setFilename**( java.lang.String s )

- **Usage**

- \* Sets the filename
- 

- *setProperty*

public void **setProperty**( java.lang.String name, java.lang.Object value )

- **Usage**

- \* Sets the value of a property

- **Parameters**

- \* **name** - the property name
    - \* **value** - the new value to set